

WestminsterResearch

<http://www.westminster.ac.uk/westminsterresearch>

**Neuro-Fuzzy Based Intelligent Approaches to Nonlinear System
Identification and Forecasting**

Alshejari, A.

This is an electronic version of a PhD thesis awarded by the University of Westminster.
© Mrs Abeer Alshejari, 2018.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch: (<http://westminsterresearch.wmin.ac.uk/>).

In case of abuse or copyright appearing without permission e-mail repository@westminster.ac.uk

Neuro-Fuzzy Based Intelligent Approaches to Nonlinear System Identification and Forecasting

By Abeer Alshejari

A Dissertation Submitted to the Graduate School of Science and
Technology in Partial Fulfillment of the Requirement for the
Degree of Doctor of Philosophy

**University of Westminster
June 2018**

ABSTRACT

Nearly three decades back nonlinear system identification consisted of several ad-hoc approaches, which were restricted to a very limited class of systems. However, with the advent of the various soft computing methodologies like neural networks and the fuzzy logic combined with optimization techniques, a wider class of systems can be handled at present. Complex systems may be of diverse characteristics and nature. These systems may be linear or nonlinear, continuous or discrete, time varying or time invariant, static or dynamic, short term or long term, central or distributed, predictable or unpredictable, ill or well defined. Neurofuzzy hybrid modelling approaches have been developed as an ideal technique for utilising linguistic values and numerical data. This Thesis is focused on the development of advanced neurofuzzy modelling architectures and their application to real case studies. Three potential requirements have been identified as desirable characteristics for such design: A model needs to have minimum number of rules; a model needs to be generic acting either as Multi-Input-Single-Output (MISO) or Multi-Input-Multi-Output (MIMO) identification model; a model needs to have a versatile nonlinear membership function.

Initially, a MIMO Adaptive Fuzzy Logic System (AFLS) model which incorporates a prototype defuzzification scheme, while utilising an efficient, compared to the Takagi–Sugeno–Kang (TSK) based systems, fuzzification layer has been developed for the detection of meat spoilage using Fourier transform infrared (FTIR) spectroscopy. The identification strategy involved not only the classification of beef fillet samples in their respective quality class (*i.e.* fresh, semi-fresh and spoiled), but also the simultaneous prediction of their associated microbiological population directly from FTIR spectra. In the case of AFLS, the number of memberships for each input variable was directly associated to the number of rules, hence, the “*curse of dimensionality*” problem was significantly reduced. Results confirmed the advantage of the proposed scheme against Adaptive Neurofuzzy Inference System (ANFIS), Multilayer Perceptron (MLP) and Partial Least Squares (PLS) techniques used in the same case study.

In the case of MISO systems, the TSK based structure, has been utilized in many neurofuzzy systems, like ANFIS. At the next stage of research, an Adaptive Fuzzy Inference Neural Network (AFINN) has been developed for the monitoring the spoilage of minced beef utilising multispectral imaging information. This model, which follows the TSK structure, incorporates a clustering pre-processing stage for the definition of fuzzy rules, while its final fuzzy rule base is determined by competitive learning. In this specific case study, AFINN model was also able to predict for the first time in the literature, the beef’s temperature directly from imaging information. Results again proved the superiority of the adopted model.

By extending the line of research and adopting specific design concepts from the previous case studies, the Asymmetric Gaussian Fuzzy Inference Neural Network (AGFINN) architecture has been developed. This architecture has been designed based on the above design principles. A clustering preprocessing scheme has been applied to minimise the number of fuzzy rules. AGFINN incorporates features from the AFLS concept, by having the same number of rules as well as fuzzy memberships. In spite of the extensive use of the standard symmetric Gaussian membership functions, AGFINN utilizes an asymmetric function acting as input linguistic node. Since the asymmetric Gaussian membership function's variability and flexibility are higher than the traditional one, it can partition the input space more effectively. AGFINN can be built either as an MISO or as an MIMO system. In the MISO case, a TSK defuzzification scheme has been implemented, while two different learning algorithms have been implemented. AGFINN has been tested on real datasets related to electricity price forecasting for the ISO New England Power Distribution System. Its performance was compared against a number of alternative models, including ANFIS, AFLS, MLP and Wavelet Neural Network (WNN), and proved to be superior. The concept of asymmetric functions proved to be a valid hypothesis and certainly it can find application to other architectures, such as in Fuzzy Wavelet Neural Network models, by designing a suitable flexible wavelet membership function. AGFINN's MIMO characteristics also make the proposed architecture suitable for a larger range of applications/problems.

Table of Contents

| | |
|--|-------------|
| ABSTRACT | II |
| TABLE OF ABBREVIATIONS | VI |
| LIST OF FIGURES | VIII |
| LIST OF TABLES | X |
| ACKNOWLEDGMENT | XI |
| 1. INTRODUCTION | 1 |
| 1.1 RATIONALE OF CURRENT RESEARCH | 1 |
| 1.2 RESEARCH AIMS AND CONTRIBUTIONS | 3 |
| 1.3 THESIS ORGANIZATION | 4 |
| 1.4 LIST OF PUBLICATIONS | 6 |
| 2. NONLINEAR SYSTEM IDENTIFICATION | 7 |
| 2.1 INTRODUCTION | 7 |
| 2.2 SYSTEM IDENTIFICATION PROCEDURES | 7 |
| 2.3 SYSTEM IDENTIFICATION MODELS CLASSIFICATION | 10 |
| 2.4 AN OVERVIEW OF SOFT COMPUTING METHODS IN SYSTEM IDENTIFICATION | 12 |
| 2.4.1 INTRODUCTION | 12 |
| 2.4.2 FUZZY SYSTEMS | 13 |
| 2.4.2.1 IDENTIFICATION WITH FUZZY MODELING | 14 |
| 2.4.3 NEURAL NETWORKS | 15 |
| 2.4.3.1 ARCHITECTURE AND LEARNING IN NEURAL NETWORKS | 16 |
| 2.5 HYBRID SOFT COMPUTING TECHNIQUES | 18 |
| 2.5.1 NEUROFUZZY SYSTEMS | 18 |
| 2.5.2 NEURAL GENETIC ALGORITHMS | 19 |
| 2.5.3 FUZZY GENETIC ALGORITHMS | 20 |
| 3. COMPUTATIONAL INTELLIGENT SYSTEM PRINCIPLES | 21 |
| 3.1 FUZZY SYSTEMS | 21 |
| 3.1.1 FUZZY SETS AND MEMBERSHIP FUNCTIONS | 22 |
| 3.1.2 FUZZY RULES AND FUZZY REASONING | 24 |
| 3.1.3 DEFUZZIFICATION METHODS | 26 |
| 3.1.3.1 THE MAMDANI INFERENCE MODEL | 26 |
| 3.1.3.2 TSK INFERENCE MODEL | 28 |
| 3.2 NEURAL NETWORK SYSTEMS | 30 |
| 3.2.1 ACTIVATION FUNCTIONS | 32 |
| 3.2.2 MULTILAYER PERCEPTRON (MLP) | 33 |
| 3.2.3 BACKPROPAGATION ALGORITHM | 34 |
| 3.2.4 WAVELET NEURAL NETWORK (WNN) | 38 |
| 3.3 NEUROFUZZY SYSTEMS | 39 |
| 3.3.1 ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) | 40 |
| 3.4 CLUSTER ANALYSIS | 44 |
| 3.4.1 FUZZY C-MEANS CLUSTERING | 45 |

| | |
|---|------------|
| 4. COMPUTATIONAL INTELLIGENCE TECHNIQUES TO ASSESS SENSORY QUALITY OF MEAT USING FTIR SPECTROSCOPY | 47 |
| 4.1 INTRODUCTION | 47 |
| 4.2 FTIR SPECTROSCOPY IN FOOD QUALITY ANALYSIS | 49 |
| 4.3 FTIR SAMPLING AND ANALYSIS | 50 |
| 4.4 ADAPTIVE FUZZY LOGIC SYSTEM (AFLS) | 54 |
| 4.4.1 AFLS ARCHITECTURE | 55 |
| 4.4.2 AFLS LEARNING PHASE PROCEDURE | 58 |
| 4.5 DATA ANALYSIS | 61 |
| 4.5.1 FIRST CASE STUDY: LOOCV TECHNIQUE | 62 |
| 4.5.2 SECOND CASE STUDY: HOLD-OUT METHOD | 71 |
| 5. COMPUTATIONAL INTELLIGENCE TECHNIQUES TO ASSESS SENSORY QUALITY OF MEAT USING MULTISPECTRAL IMAGING | 77 |
| 5.1 INTRODUCTION | 77 |
| 5.2 MULTISPECTRAL IMAGING APPROACHES IN FOOD QUALITY ANALYSIS | 79 |
| 5.3 MULTISPECTRAL IMAGING - SAMPLING AND ANALYSIS | 79 |
| 5.4 AFINN ARCHITECTURE | 85 |
| 5.4.1 CLUSTERING ALGORITHM | 86 |
| 5.4.2 FEED-FORWARD ANALYSIS OF AFINN | 87 |
| 5.4.3 TUNING PREMISE AND CONSEQUENCE AFINN PARAMETERS | 89 |
| 5.5 DATA ANALYSIS | 90 |
| 5.5.1 CLASSIFICATION OF MEAT SAMPLES | 90 |
| 5.5.2 TEMPERATURE IDENTIFICATION MODEL | 91 |
| 5.5.3 TOTAL VIABLE COUNTS IDENTIFICATION MODEL | 93 |
| 5.5.3.1 AIR CASE | 93 |
| 5.5.3.2 MAP CASE | 97 |
| 5.5.4 SALMONELLA IDENTIFICATION MODEL | 101 |
| 6. SHORT-TERM ELECTRICITY PRICE FORECASTING USING ASYMMETRIC FUZZY NEURAL NETWORKS | 104 |
| 6.1 INTRODUCTION | 104 |
| 6.2 THE STPF CASE PROBLEM | 108 |
| 6.3 ASYMMETRIC NEUROFUZZY MODEL (AGFINN) | 111 |
| 6.3.1 FCM CLUSTERING ALGORITHM | 112 |
| 6.3.2 FEED-FORWARD ANALYSIS OF AGFINN | 113 |
| 6.3.3 AGFINN LEARNING PHASE | 116 |
| 6.3.3.1 AGFINN-TSK HYBRID LEARNING SCHEME | 118 |
| 6.4 RESULTS AND DISCUSSION | 120 |
| 6.4.1 CASE STUDY 1 | 122 |
| 6.4.2 CASE STUDY 2 | 126 |
| 6.4.3 CASE STUDY 3 | 129 |
| 7. CONCLUSIONS AND FUTURE ENHANCEMENTS | 135 |
| 7.1 CONCLUSIONS | 135 |
| 7.2 FUTURE ENHANCEMENTS | 137 |
| REFERENCES | 138 |

TABLE OF ABBREVIATIONS

| | |
|----------------------|--|
| A_r | Accuracy Factor |
| AIR | Aerobic |
| AFINN | Adaptive Fuzzy Inference Neural Network |
| AFLS | Adaptive Fuzzy Logic System |
| AGFINN | Asymmetric Gaussian Fuzzy Inference Neural Network |
| AGFINN-CA | CA-based AGFINN |
| AGFINN-TSK | TSK-based AGFINN |
| ANFIS | Adaptive Neuro Fuzzy Inference System |
| ANN | Artificial Neural Networks |
| APC | Aerobic Plate Count |
| APE | Absolute Percentage Error |
| AR | Auto Regressive |
| ARMA | Auto Regressive Moving Average |
| ARMAX | Auto Regressive Moving Average exogenous model |
| B_r | Bias Factor |
| BP | Backpropagation |
| CA | Center Average |
| CI | Computational Intelligence |
| DFA | Discriminant Function Analysis |
| EC | Evolutionary Computation |
| EKF | Extended Kalman Filter |
| FALCON | Fuzzy Adaptive learning Control Network |
| FCM | Fuzzy C-Means |
| FGA | Fuzzy Genetic Algorithm |
| FIS | Fuzzy Inference System |
| FKB | Fuzzy Knowledge Base |
| FL | Fuzzy Logic |
| FLS | Fuzzy Logic System |
| FTIR | Fourier Transform Infrared Spectroscopy |
| GA | Genetic Algorithm |
| GD | Gradient Descent |
| ISO | Independent System Operator |
| ISONE | ISO-New England |
| LOOCV | Leave-One-Out Cross-Validation |
| LS | Least Square |
| LSSVM | Least Square SVM |
| LTPF | Long-Term Price Forecasting |
| LVQ | Learning Vector Quantization |
| MAE | Mean Absolute Error |
| MAP | Modified Atmosphere |
| MAPE | Mean Absolute Percentage Error |
| MAPR | Mean Absolute Percentage Residual |
| MCP | Market Clearing Price |
| MF | Membership Function |
| MIMO | Multi-Input Multi-Output |
| MIMO-CA | CA-based MIMO |
| MISO | Multi-Input-Single-Output |
| MISO-TSK | TSK-based MISO |

| | |
|----------------------|---|
| MLP | Multi Layer Perceptron |
| MRPR | Mean Relative Percentage Residual |
| MSE | Mean Squared Error |
| MTPF | Medium-Term Price Forecasting |
| MW | Megawatts |
| MWh | Megawatt-hours |
| NARX | Nonlinear Autoregressive exogenous model |
| NARMAX | Nonlinear Autoregressive Moving Average exogenous model |
| NF | Neuro Fuzzy |
| NGA | Neural Genetic Algorithm |
| NIR | Near Infrared |
| NIPLAS | Nonlinear Iterative Partial Least Squares |
| NLR | Non Linear Regression |
| NN | Neural Network |
| PCA | Principal Component Analysis |
| PLS | Partial Least Square |
| R² | Coefficient of determination |
| RBF | Radial Basis Functions |
| RLS | Recursive Least Square |
| RMSE | Root Mean Squared Error |
| RPCA | Robust Principal Component Analysis |
| ROI | Region of Interest |
| SEP | Standard Error of Prediction |
| SC | Soft Computing |
| SI | System Identification |
| STPF | Short-Term Price Forecasting |
| SVM | Support Vector Machine |
| TSK | Takagi–Sugeno–Kang |
| TVC | Total Viable Count |
| U₁ | Theil U1 |
| WNN | Wavelet Neural Network |

LIST OF FIGURES

| | | |
|------------------|--|----|
| Fig. 2.1 | A diagram of the salient steps in system identification | 8 |
| Fig. 2.2 | Modeling approaches ranging from clear white to black box | 11 |
| Fig. 2.3 | Productivity membership functions | 14 |
| Fig. 2.4 | Feedforward NN | 17 |
| Fig. 2.5 | Recurrent NN | 17 |
| Fig. 3.1 | Classes of membership functions | 23 |
| Fig. 3.2 | Fuzzy Inference System structure | 25 |
| Fig. 3.3 | The original Mamdani FIS using (max-min) composition | 27 |
| Fig. 3.4 | Various defuzzification methods | 27 |
| Fig. 3.5 | TSK fuzzy model | 29 |
| Fig. 3.6 | Single perceptron process | 31 |
| Fig. 3.7 | Neuron activation functions | 33 |
| Fig. 3.8 | MLP Neural Network structure | 34 |
| Fig. 3.9 | (a) WNN structure (b) Morlet Wavelet basis function | 38 |
| Fig. 3.10 | A two-input ANFIS architecture | 40 |
| Fig. 3.11 | A Comparison between Hard and Fuzzy clustering | 45 |
| Fig. 4.1 | FTIR spectroscopy device / principles | 51 |
| Fig. 4.2 | FTIR spectra collected from beef samples stored at 10°C | 52 |
| Fig. 4.3 | Population dynamics of TVC at various temperatures for beef samples | 53 |
| Fig. 4.4 | A 3-D plot for the first three principal components | 54 |
| Fig. 4.5 | AFLS architecture | 55 |
| Fig. 4.6 | Triangular shape membership function | 57 |
| Fig. 4.7 | Consequent fuzzy set placed on mass-less beam | 58 |
| Fig. 4.8 | AFLS (a) performance for TVCs, and (b) % relative errors (LOOCV case) | 63 |
| Fig. 4.9 | ANFIS (a) performance for TVCs, and (b) % relative errors (LOOCV case) | 66 |
| Fig. 4.10 | MLP (a) performance for TVCs, and (b) % relative errors (LOOCV case) | 68 |
| Fig. 4.11 | PLS performance for TVCs | 69 |
| Fig. 4.12 | AFLS (a) performance for TVCs, and (b) % relative errors (Case 2) | 72 |
| Fig. 4.13 | ANFIS (a) performance for TVCs, and (b) % relative errors (Case 2) | 73 |
| Fig. 4.14 | MLP (a) performance for TVCs, and (b) % relative errors (Case 2) | 75 |
| Fig. 5.1 | Schematic of the proposed data analysis | 77 |
| Fig. 5.2 | Structure of proposed decision support system | 78 |
| Fig. 5.3 | Example of Aerobic vs. MAP storage | 80 |
| Fig. 5.4 | Population dynamics of TVC and XLD at various temperatures for minced beef samples | 81 |

| | | |
|------------------|--|-----|
| Fig. 5.5 | Multispectral Imaging Concept | 82 |
| Fig. 5.6 | Videometer Sensorial System | 82 |
| Fig. 5.7 | Selected spectra for both AIR and MAP cases | 83 |
| Fig. 5.8 | Structure of AFINN system | 86 |
| Fig. 5.9 | Clustering concept | 87 |
| Fig. 5.10 | AFINN prediction model for TVC (AIR case- all inputs) | 93 |
| Fig. 5.11 | AFINN prediction model for TVC (AIR case- RPCA inputs) | 94 |
| Fig. 5.12 | AFINN prediction model for TVC (MAP case- all inputs) | 98 |
| Fig. 5.13 | AFINN prediction model for TVC (MAP case- RPCA inputs) | 100 |
| Fig. 5.14 | AFINN prediction model for XLD (AIR case- all inputs) | 101 |
| Fig. 5.15 | AFINN prediction model for XLD (MAP case- all inputs) | 102 |
| Fig. 6.1 | ISONE Electricity price times- series corresponding to 4h | 110 |
| Fig. 6.2 | ISONE Electricity price times- series corresponding to 22h | 110 |
| Fig. 6.3 | Structure of AGFINN-CA system | 111 |
| Fig. 6.4 | Structure of AGFINN-TSK system | 112 |
| Fig. 6.5 | Structure of Asymmetric MF | 114 |
| Fig. 6.6 | Forecasting for Electricity Price at 22h, (Case Study 1) | 123 |
| Fig. 6.7 | Forecasting for Electricity Price at 4h, (Case Study 1) | 123 |
| Fig. 6.8 | Comparison of AGFINN-based models (Case Study 1) | 125 |
| Fig. 6.9 | Forecasting for Electricity Price at 4h, (Case Study 2) | 127 |
| Fig. 6.10 | Forecasting for Electricity Price at 22h, (Case Study 2) | 127 |
| Fig. 6.11 | Comparison of AGFINN-based models (Case Study 2) | 129 |
| Fig. 6.12 | Forecasting for Electricity Price at 4h, (Case Study 3) | 130 |
| Fig. 6.13 | Forecasting for Electricity Price at 22h, (Case Study 2) | 130 |
| Fig. 6.14 | Comparison of AGFINN-based models (Case Study 3) | 132 |
| Fig. 6.15 | AGFINN-RLS Forecasting performance (Case Study 3) | 133 |

LIST OF TABLES

| | | |
|-------------------|---|-----|
| Table 2.1 | A comparison between Fuzzy Systems and Neural Networks | 19 |
| Table 4.1 | Eigenvalues and proportion of the first seven PCs for PCA scheme | 53 |
| Table 4.2 | Confusion Matrix for AFLS acting as classifier – LOOCV case | 62 |
| Table 4.3 | Performance of AFLS - LOOCV case | 65 |
| Table 4.4 | Performance of ANFIS - LOOCV case | 67 |
| Table 4.5 | Confusion Matrix for MLP acting as classifier – LOOCV case | 67 |
| Table 4.6 | Performance of MLP - LOOCV case | 69 |
| Table 4.7 | Performance of PLS - LOOCV case | 70 |
| Table 4.8 | Confusion Matrix for AFLS acting as classifier – Case 2 | 71 |
| Table 4.9 | Performance for AFLS - Case 2 | 72 |
| Table 4.10 | Performance for ANFIS - Case 2 | 74 |
| Table 4.11 | Confusion Matrix for MLP acting as classifier – Case 2 | 74 |
| Table 4.12 | Performance for MLP - Case 2 | 75 |
| Table 5.1 | Robust PCA scheme | 84 |
| Table 5.2 | Confusion matrix for class of storage conditions | 91 |
| Table 5.3 | Confusion matrix for temperature using AFINN model | 92 |
| Table 5.4 | Confusion matrix for temperature using ANFIS / MLP models | 93 |
| Table 5.5 | Statistical performance for AIR case (all inputs) | 94 |
| Table 5.6 | Statistical performance for AIR case (all inputs- comparison) | 95 |
| Table 5.7 | Statistical performance for AIR case (PCA inputs) | 97 |
| Table 5.8 | Statistical performance for MAP case (all inputs) | 98 |
| Table 5.9 | Statistical performance for MAP case (all inputs- comparison) | 99 |
| Table 5.10 | Statistical performance for MAP case (PCA inputs) | 101 |
| Table 5.11 | Statistical performance for AIR case (XLD case) | 102 |
| Table 5.12 | Statistical performance for MAP case (XLD case) | 103 |
| Table 6.1 | Statistical indices of the ISONE dataset (2006-2007) | 109 |
| Table 6.2 | Statistical performance for Electricity Price Forecasting Models at 4h, | 124 |
| Table 6.3 | Statistical performance for ElectricityPrice Forecasting Models at 22h, | 124 |
| Table 6.4 | Statistical performance for Electricity Price Forecasting Models at 4h, | 128 |
| Table 6.5 | Statistical performance for ElectricityPrice Forecasting Models at 22h, | 128 |
| Table 6.6 | Statistical performance for Electricity Price Forecasting Models at 4h, | 131 |
| Table 6.7 | Statistical Performance for ElectricityPrice Forecasting Models at 22h, | 131 |

ACKNOWLEDGMENT

I would like to express my appreciation and heartfelt thanks to all those who have been a source of support in this academic journey.

To Dr. Vassilis Kodogiannis, my director of studies, I would like to extend my sincerest gratitude for your unwavering support and guidance throughout this project. It was a gratifying experience to be mentored by you. Your invaluable insight and expertise shaped my research, and for that I am grateful.

To Dr. Andrzej Tarczynski, I would like to extend my thanks for taking the time to advise me in my research, and for your continuous assistance and feedback when needed.

To my family, thank you for your constant support and encouragement throughout this experience.

Chapter One

INTRODUCTION

1.1 Rationale of Current Research

System modeling, in its general definition, is practically considered as an adequate representation of how variables interact and relate to each other to produce an acceptable output. Among all the tasks in data analysis, modeling is considered to be the most frequently occurring process. The identification/modeling of nonlinear systems has always been a challenging problem due to the unique nature and dissimilar characteristics for each system. A fundamental goal in the process of nonlinear system modeling is to produce a universal model, which is capable of identifying a wide class of different structured systems. Hence, a vast range of techniques to address these tasks has been developed, which vary from fundamental mathematical forms to more sophisticated artificial intelligence approaches.

For many years there have been a number of mathematical techniques designed to facilitate this modelling task. These range greatly in complexity from very simple models capable of operating general systems like thermostats in heating systems to more complex ones capable of tracking/predicting complex dynamic systems and recognising complex objects in a variety of surroundings. Conventional modelling methods based around a linear sum of a given set of factors have performed remarkably well. In fact, as long as the real world remains fairly close to the desired observed states around which the parameters were obtained they are often sufficient. However for dynamic systems that change even a little from their known state or when asked to distinguish between subjects, which are not linearly separable, different techniques are required.

For the purpose of data analysis, Soft Computing methods (SC) have experienced high popularity due to their ability to solve complicated systems, particularly where mathematical approaches proved to be insufficient. Modeling real-world problems is often a challenge as one usually faces difficult and ill-defined systems. Exact mathematical models in such cases tend to be unfeasible, impractical or come with high computational cost. Soft Computing methods that have been developed from emulating intelligent phenomenon in human and nature provide a suitable framework to deal with such complications. SC involves methodologies such as neural networks (NN), fuzzy logic (FL) and Evolutionary Computation (EC), and is focused on the study of adaptive mechanism to assist intelligent behavior in inconsistent and imprecise environments.

Artificial Neural Networks originated as an attempt to mimic the structure of the biological brain. They are comprised of a number of simple processing elements which when taken together possess massive representational capabilities. Multilayer Perceptron (MLP), radial basis function networks, Recurrent Neural Networks (RNN) and Backpropagation (BP) learning algorithm with Gradient Decent method (GD) are usually employed to approximate the input/output mapping of nonlinear systems. Their main problems are that it is a time-consuming procedure and the learnt network may not be optimal due to the lack of knowledge to select the proper network structure. Fuzzy systems on the other hand, accept numeric inputs and convert them into linguistic values that can be manipulated with linguistic IF-THEN rules and with fuzzy logic operations, such as fuzzy implication and composition rules of inference. However, at present there is no systematic procedure for the design of a fuzzy system. Usually the fuzzy rules are generated by converting human operators' experience into fuzzy linguistic form directly and by summarizing the system behaviour (sampled input-output pairs) of the operators. However, it is rather difficult to obtain adequate fuzzy rules and membership functions because these are most likely to be influenced by the intuitiveness of the operators and the expert designers.

Neurofuzzy (NF) hybrid modeling approaches are a set of methods created by combining Neural Networks and Fuzzy Logic. Such hybrid approach is supported by the desired qualities of both the fuzzy logic and the neural network methods hence generating adaptable models, which could be represented and interpreted linguistically by fuzzy rules structure while learning from experimental data. In dynamic system identification, neurofuzzy systems incorporating a Takagi-Sugeno-Kang (TSK) scheme possess a very good interpretation, which is superior to most, if not all, alternative defuzzification approaches. However, TSK-based fuzzy systems may require a huge number of rules and associated coefficients in order to achieve the desired accuracy. A large network structure is associated with an undesirably high computational cost. The Adaptive Neurofuzzy Inference System (ANFIS) architecture is a classic representative of TSK-based models.

This thesis investigates the ability of advanced neurofuzzy soft computing approaches to learn how to identify adequately complex nonlinear systems. Two important applications areas have been targeted, i.e. energy forecasting and prediction of bacteria in food. One important, and still open, problem in NF systems is to how to determine their initial structure, including the number of fuzzy rules, the initial parameters of membership functions of the premise and consequent parts in each rule. Existing NF systems, especially those that follow the TSK-structure, suffer from the co-called "*curse of dimensionality*" problem, which practically is related to the excessive number of generated fuzzy rules. In addition, although many NF systems have been developed for a number of applications, the area of food microbiology is still considered as "*Terra incognita*" for such advanced systems.

1.2 Research Aims and Contributions

The main objective of this thesis is the development of novel neurofuzzy architecture, which is capable of tackling highly nonlinear problems. Knowing the strengths as well as the deficiencies of existing NF systems, the following sub-objectives have been considered:

- A NF model needs to have minimum possible number of rules
- A NF model needs to be generic acting either as MISO or MIMO model
- A NF model needs to have a versatile nonlinear membership function.

In this research, a hierarchical approach has been followed, towards the development of a general purpose advanced NF identification model. At each stage of research, valuable experience has been gained and such experience was accumulated to the final model development. A number of advanced NF models have been developed for two important applications: Meat Spoilage Detection and Electricity price Forecasting.

Initially, an Adaptive Fuzzy Logic System (AFLS) model has been developed for the detection of meat spoilage utilising Fourier transform infrared (FTIR) sensorial information. In AFLS models, the number of memberships for each input variable is directly associated to the number of rules, thus the “*curse of dimensionality*” problem is significantly reduced. However, the number of rules is still determined by the user. Currently, in the area of food microbiology, either classic multivariate techniques or simple MLP networks are used as identification models. In this research, for the first time in literature, a MIMO NF system has been applied successfully simultaneously as a classifier of beef samples in their respective quality class (i.e. fresh, semi-fresh and spoiled), as well as a predictor of their associated microbiological population.

In a further stage of research, an Adaptive Fuzzy Inference Neural Network (AFINN) model has been developed for the detection of meat spoilage utilising multispectral sensorial information. AFINN follows the classic MISO TSK-based structure. However, a clustering pre-processing algorithm is applied for the sample data in order to organize feature vectors into clusters, such that points within a cluster are closer to each other than vectors belonging to different clusters. AFINN provides a means of controlling the growth of the number of local linear systems when the order of the system under consideration increases, so that least-squares estimation can be applied without performance degradation. The idea of clustering pre-processing stage is considered as important, as it can determine the number of fuzzy rules as well as the initial values for the related fuzzy memberships in the fuzzification layer. The application of NF systems in this domain is again novel whereas the identification of temperature based only on multispectral imaging information is considered to be truly innovative.

Results from AFLS and AFINN schemes are compared against models based on ANFIS, MLP, and Partial Least Squares (PLS). Such comparison is considered as an essential practice, as we have to emphasise the need of induction to the area of food microbiology, advanced learning-based modelling schemes, which may have a significant potential for the rapid and accurate assessment of meat spoilage. Such an accurate assessment/prediction could allow a more efficient management of products in the food chain.

All these ideas have been also incorporated to the final developed Asymmetric Gaussian Fuzzy Inference Neural Network (AGFINN) model, which additionally utilise an asymmetric Gaussian function. AGFINN has been tested on real datasets related to electricity price forecasting. AGFINN concept has been built either as an MISO or MIMO system and utilizes also a hybrid-learning algorithm. AGFINN's MIMO characteristics also make the proposed architecture suitable for a large range of applications/problems. Results from AGFINN schemes are compared against models based on AFLS, ANFIS, MLP and Wavelet Neural Networks (WNN). In general, NF systems are considered to be more accurate than single layer networks, like MLP or even WNNs. In this research, it has been proved that in some cases, WNN could outperform ANFIS performance. The importance/utilisation of wavelet functions as potential “fuzzy” membership functions is something that needs to be considered carefully.

1.3 Thesis Organization

The thesis is organised into seven chapters. Following this introductory chapter the next chapter, Chapter 2, gives a summary of nonlinear system identification with models procedures and classification. Then an overview of soft computing methods used in system identification is introduced. In addition, several architectures of hybrid techniques are explained with an emphasis of the importance of such integration between soft computing methods. Chapter 3 provides an analysis of specific soft computing paradigms with necessary mathematical details. It starts with fuzzy systems and its concepts. Then, neural networks with training learning algorithms have been discussed. Hybrid wavelet neural network principles are also explained. Finally, a brief introduction to cluster analysis is provided, with emphasis on the Fuzzy C-means method.

Chapter 4 introduces the Adaptive Fuzzy Logic System (AFLS) architecture and its application to the rapid and non-destructive detection of meat spoilage microorganisms during aerobic storage at chill and abuse temperatures, utilising FTIR spectroscopy. Datasets related to FTIR spectra and the correlated microbiological analysis (i.e. total viable counts – TVC) from beef fillets, were provided by the Agricultural University of Athens, and the specific application is considered to be as the first time that NF systems are applied to this

specific important problem. The identification strategy involved not only the classification of beef samples in their respective quality class (i.e. fresh, semi-fresh and spoiled), but also the prediction of their associated microbiological population directly from FTIR spectra. The realisation of this strategy has been fulfilled with the development of a MIMO neurofuzzy model which incorporates a prototype defuzzification scheme, while utilising an efficient, compared to TSK-systems, fuzzification layer. In the case of AFLS, the number of memberships for each input variable is directly associated to the number of rules, hence, the “curse of dimensionality” problem is significantly reduced. The performance of the proposed scheme is then compared against ANFIS, in terms of prediction accuracy and structure simplicity. Subsequent comparison against the MLP and PLS had also been made.

Chapter 5 introduces the Adaptive Fuzzy Inference Neural Network (AFINN), an TSK based MISO structure, and its application to associate, for the first time according to literature, spectral data acquired by multispectral imaging techniques with meat spoilage. AFINN is a NF that incorporates a clustering pre-processing stage for the definition of fuzzy rules, while its final fuzzy rule base is determined by competitive learning. In addition, a hybrid-learning rule is applied for the tuning network’s parameters. An intelligent decision support system based on AFINN, initially classifies minced beef samples stored either aerobically or under modified atmosphere packaging and then predicts for each case, the total viable counts of bacteria. The innovation of the proposed approach is further extended to the identification of the temperature used for storage, utilizing only imaging spectral information. Results from AFINN scheme are compared against models based on ANFIS, MLP, as well as non-linear and linear PLS regression schemes.

In chapter 6, a novel asymmetric neurofuzzy network AGFINN architecture has been presented, which incorporates the experience gained from AFLS and AFINN NF models. The proposed scheme has been developed utilising two alternative defuzzification schemes acting either as an MIMO or MISO configuration. The Fuzzy C-Means clustering has been used as a pre-processing step to derive the required number of clusters and eventually the number of fuzzy rules in AGFINN model, while an asymmetric function, acting as input linguistic node, has replaced the standard symmetric Gaussian membership function usually appeared in neurofuzzy models. The AGFINN model that utilise the Takagi–Sugeno–Kang defuzzification method utilises also a hybrid-learning algorithm, which include Gradient Descent (GD) and Recursive Least Squares (RLS). AGFINN has been developed for short-term price forecasting of the electricity prices in ISO New England market and its performance clearly outperforms that of AFLS, ANFIS, WNN and MLP systems.

Finally, chapter 7 draws conclusions and possible directions for future work. Three recommended enhancements, which were out of the scope of this research, are presented here.

1.4 List of Publications

Referred Journal Papers

1. V.S. Kodogiannis, A. Alshejari, “An Adaptive Neuro-Fuzzy Identification Model for the Detection of Meat Spoilage”, *Applied Soft Computing*, Elsevier, Vol. 23, pp. 483-497, 2014, ISSN: 1568-4946, Impact Factor (IF): 3.541.
2. A. Alshejari, V.S. Kodogiannis, “An intelligent decision support system for the detection of meat spoilage using multispectral images”, *Neural Computing and Applications*, Springer, pp. 1-18, 2016 (Online), IF: 2.505.
3. V.S. Kodogiannis, A. Alshejari, “Short-term Electricity price forecasting using Asymmetric Fuzzy Neural Networks”, submitted to *Electric Power Systems Research*, Elsevier, Aug. 2017, IF: 2.688.

Referred Conferences

1. A. Alshejari, V. Kodogiannis, I. Petrounias, “An Adaptive Neuro-Fuzzy Model for the Detection of Meat Spoilage using Multispectral Images”, 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) Istanbul, Turkey Aug 2015, pp. 1-7, **ISBN: 978-1-4673-7428-6**
2. Alshejari, A., Kodogiannis, V., Petrounias I, “A Rapid Detection of Meat Spoilage using FTIR and Neuro-fuzzy systems”, SAI Intelligent Systems Conference 2015 London Nov 2015, pp. 576-584, 2015, **ISBN: 978-1-4673-7606-8**.
3. Kodogiannis, V. and Alshejari, A., “A Fuzzy-Wavelet Neural Network Model for the Detection of Meat Spoilage using an Electronic Nose”, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Jul 2016, Vancouver, BC, Canada, pp. 710 - 717, 2016.
4. Kodogiannis, V. and Alshejari, A., “Neuro-Fuzzy based Identification of Meat Spoilage using an Electronic Nose”, 8th IEEE International Conference on Intelligent Systems IS'16, Sep 2016, Sofia, Bulgaria, pp. 96-103, 2016
5. Alshejari, A. and Kodogiannis, V., “Day ahead hourly Price Forecast in ISO New England Market using Neuro-Fuzzy Systems”, Intelligent Systems Conference 2017, Sep 2017, London, UK.
6. A Alshejari, V Kodogiannis, “Electricity Price Forecasting using Asymmetric Fuzzy Neural Network Systems”, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Jul 2017, Italy, 2017.

Chapter Two

NONLINEAR SYSTEM IDENTIFICATION

2.1 Introduction

A mathematical form that describes the static or dynamic behavior of any system is of significant requirement to improve our understanding of numerous real world problems. A number of complex systems are encountered in technical areas such as electrical engineering, as well as the natural sciences and non-technical problems generally found in physics, biology, medicine and economics.

The process of modeling of a system is to describe the temporal behavior of every process by characterizing the system conditions and parameters through the development of mathematical abstractions. Two distinct approaches are followed in model development; theoretical path using the first-principles of a system and empirical approach based on the analysis of experimental input-output data. The latter is called System Identification (SI) and is considered to be the natural and practical alternative for modeling complex systems in particular, since the fundamental level information is either too difficult to be tackled or partially unknown. Moreover, a key asset of SI is the potential of capturing detailed information from the observed data that could otherwise be overlooked in theoretical modeling [1].

2.2 System Identification Procedures

System identification is the process of utilizing observed input-output data in the development of a mathematical model. Hence, given the data, system identification is a mapping between the input domain (causes) and the output domain (effects). System identification combines the knowledge of the model developer with their creativity to design the most consistent and accurate representation of the system. Fig. 2.1 illustrates the schematic of the salient steps in system identification. The identification employment is divided into 4 main steps [2,3]:

I. Data Acquisition

The integral step of creating an identification system model is data acquisition. It is the initial step and one that establishes the foundation by which the model is generated and evaluated.

II. Data Pre-Processing

Upon acquiring the data, it is often in its initial form that is usually inadequate for use in the model. In turn, a pre-processing procedure is vital to ensure the data is in an appropriate form. This step also accounts for certain features that influence the quality of raw data, namely the problem of outliers among others. Outliers refer to anomalous data points that do not appear to correspond to or match the rest of the data. Although these pose a challenge to the model due to the difficulty of identifying and tackling them, a number of statistical methods can be used in this instance to assist in this preprocessing step.

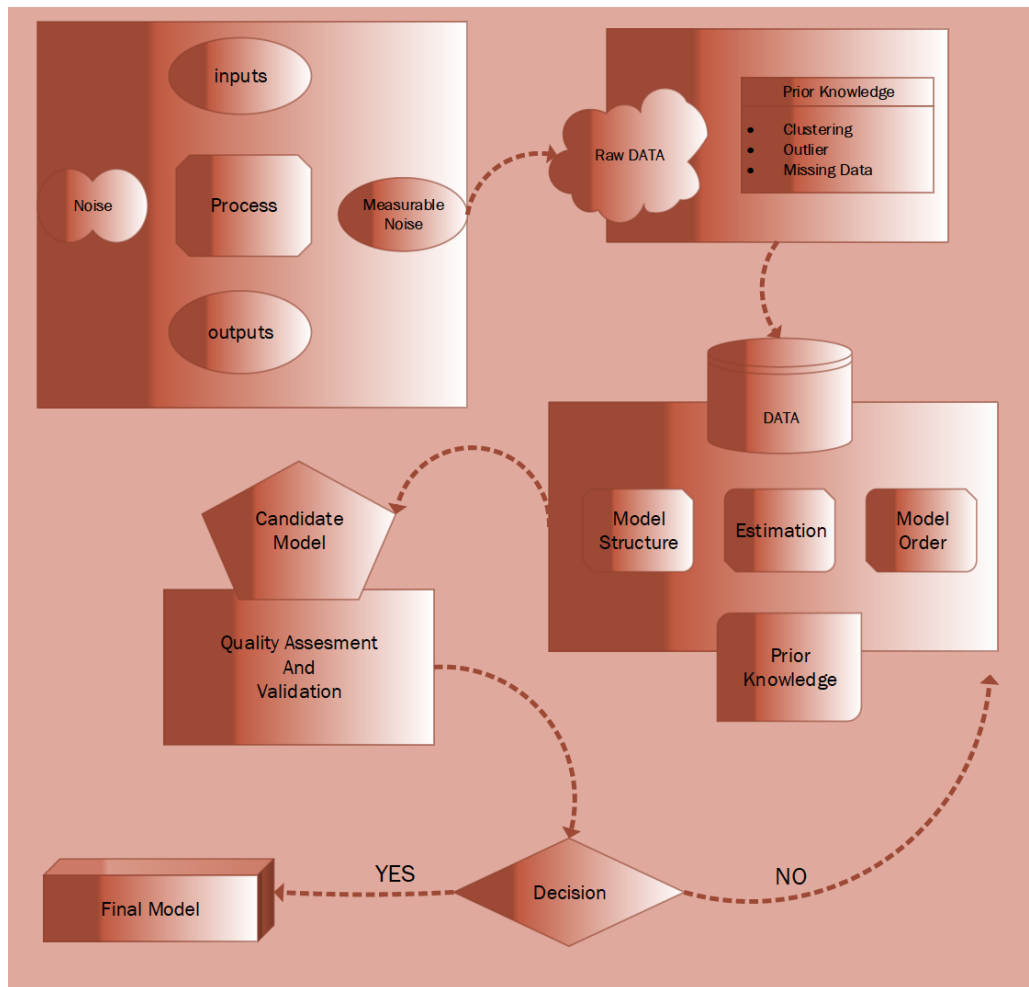


Fig. 2.1 A diagram of the salient steps in system identification

It has to be emphasized that this step is where all assumptions and requirements of developing the system identification model can be accounted for and implemented. This includes any available a priori knowledge regarding the structure and order of the model and the set of parameters. Consequently, the incorporation of such information leads to a model development process with higher transparency. The modelling process usually begins with an initial choice of a model with preliminary specification in terms of structure and type.

Any blind guess is likely to lead to a twisty route with a divergent result, and a decent guess should be based on a priori information. An introductory investigation of the data such as clustering is certainly considered as beneficial knowledge to be incorporated to reflect the process behaviour. Additionally, methods such as dimensionality reduction and filtering could be proved to be beneficial preprocessing tools.

III. Model Development

This is the fundamental step of the identification procedure: model development. Such step is where the model order identified, the structure formulated, and the parameters estimated. It is perhaps the most challenging and time-consuming stage of the procedure. There is a range of candidate models that can be chosen in this step, depending on the desired requirements. One consideration is the prior knowledge available on the type and structure of the model. Once a selection of a candidate model is confirmed, the problem becomes an optimization case in order to modify the model factors. The target function to be minimised is usually based on a metric space such as the square error between the predicted and actual values. Overall, this step tends to be initiated by an approximation of the model's configuration, which is followed by refinements along the way.

VI. Model Evaluation

This step comprises of assessing and validating the quality of the resulting identification model. What is essentially judged is the model's capability of effectively describing disparity in the output of the training data, in anticipation of minimal errors in prediction given a sufficient level of precision.

Statistical residuals error evaluation as well as an analysis of model fitness is performed simultaneously. Based on the findings, the decision is made for the amount of refinements needed on previous stages. The overall aim is to achieve an acceptable point of balance between the prediction accuracy and precision of the variance values. Shortcomings could result from an unsatisfactory model structure, order or data quality. Hence, any refinements should be based on the diagnostic feedback provided by the quality assessment.

Unlike its theoretical counterpart, empirical modeling provides the developer with a certain level of flexibility and freedom in choosing the structure of the model. However, this advantage also has its downfalls, particularly with respect to the accuracy of mathematical description and the problem of overfitting. The problem of overfitting occurs when the model is trained to apprehend the local characters of the data rather than the global features. The attempt to decrease the bias of the prediction occurs with the risk of high variance errors in the parameter approximations. This leads to poor prediction ability and might in extreme situations end up into an unbounded model. The ideal model would both accurately capture

the local features of the experimented input-output effects and generalize the model ability of identifying a fresh data set. This systematic approach can be expected to lead to a working model in the least number of iterations, subject to the learning algorithm. A useful and well-qualified model necessarily requires a sound knowledge of the impact of the initial choices and the preceding decisions made throughout the identification process. It's worth mentioning that system identification is a journey, which designs a well made and suitable model that is considered as the centerpiece of science and art.

2.3 System Identification Models Classification

Further categories can be distinguished within the two distinct approaches of first-principle and empirical modeling. The following classifications are based on different criteria found in the system.

- **Response features: Static vs. dynamic systems**

A key difference between static and dynamic models is that the latter integrates time into the explanatory variables, whereas the former is time-independent. Naturally, dynamic models are suitable for applications where the time variable is fundamental to understand the behaviour of the system, such as financial forecasting and control applications. On the other hand, one of the most important applications to the static field is pattern recognition.

- **System nature assumption: Linear vs. non-linear systems**

A linear system follows the principles of superposition and homogeneity [4] as shown in the following definition:

Definition 2.1

A system with input $u[t]$ and output $y[t]$ where $y[t] = T\{u[t]\}$ is linear if and only if

$$\begin{aligned} \text{I.} \quad & T\{\alpha u[t]\} = \alpha T\{u[t]\} \\ \text{II.} \quad & T\{\alpha_1 u_1[t] + \alpha_2 u_2[t]\} = \alpha_1 T\{u_1[t]\} + \alpha_2 T\{u_2[t]\} \end{aligned}$$

Where T is the transformation operator, and α is a constant.

In contrast, nonlinear systems do not obey the mentioned linear principles and their behavior shows unpredictable trace. Most real world problems are nonlinear with inherent nature. The system is described by a set of equations that include one or more nonlinear functions in term of the explanatory variables. Linear techniques are commonly applied to solve the system of equations, however many studies proved them to be inadequate in approximating systems with strong nonlinear behaviors.

- **Structure of the model: Parametric vs. Non-Parametric Models**

Parametric model assumes that a finite number of parameters are directly related to the objective qualities of the system, and are adequate to model any process. A specific difference or differential equation commonly forms the structure of the required mathematical model. Conversely, in non-parametric models, fewer structuring rules are imposed on the model. Rather, these models provide an implicit relation between the inputs and their corresponding response. A large number of factors are used to identify the non-parametric model however minimal initial knowledge of the system is required for the estimation [2].

- **Existing knowledge: Grey shades palette**

In the classification of models, we can generate a palette of grey shades, each attributed to models with certain features. White-box models are the product of first-principle modeling, which is based on the direct knowledge of physical laws characterizing the system. This approach guarantees a high level of transparency in the final model that is not present in empirical models. This transparency depends on the knowledge imposed, thus empirical models tend to lie on the cloudier, grey side of the spectrum. On the other hand, a purely empirical approach based solely on experimental measurements are labeled as black-box models. One well-established example of a black-box model is neural networks.

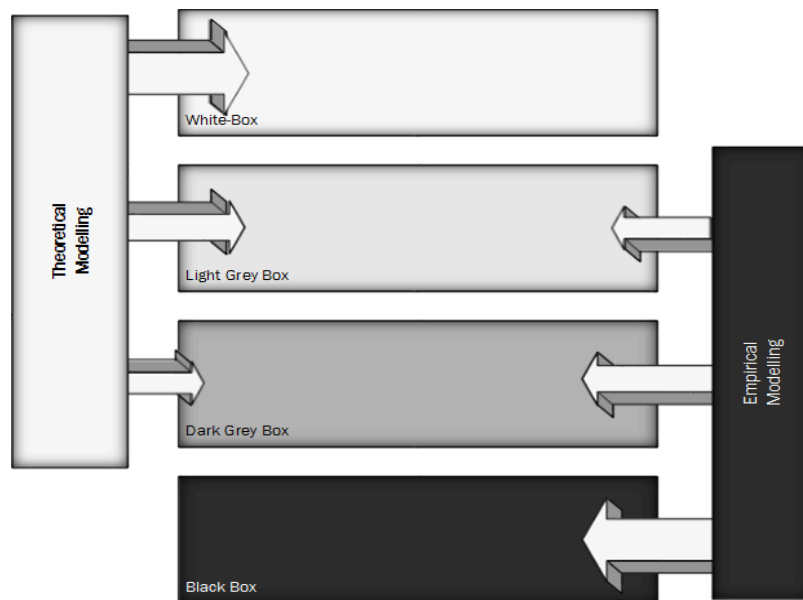


Fig. 2.2 Modeling approaches ranging from clear white to black box

Fig. 2.2 illustrates how increasing the knowledge of physical laws in the system decreases the opaqueness of the model, which generates this palette of grey shades of modeling techniques. The size of the arrows indicates the amount of knowledge inserted in each box [4].

2.4 An Overview of Soft Computing Methods in System Identification

2.4.1 Introduction

As many processes in control, monitoring and prediction systems reveal a nonlinear static and dynamic behavior, they are naturally more challenging to be identified and modelled. Many researchers had proven the deficiency of techniques based on conventional analysis where the rigorous mathematical equations had failed to accurately describe the nonlinear behavior in dynamic systems. Consequently, approaches based on classical methods do not provide an appropriate conceptual framework for dealing with the imprecise and complex nature of real world problems especially if wide domain of functionality is considered.

The general model structure to describe linear process is assumed to consist of a series of inputs and an output. The measure response in any process includes two different components, deterministic component (u) associated with the known inputs, and the stochastic white noise components (v) related to disturbance, unmeasured data and modeling error. The two distinct influences on the system could be characterized using two linear filters at time k . A linear filter $G(q)$ called the Input Transfer Function is applied for modelling the deterministic part $u(k)$, while the Noise Transfer Function $H(q)$ is used to relate the stochastic noise part $v(k)$ to the output $y(k)$. Both linear filters could be split into their numerator and denominator components with a shared denominator factor $A(q)$ where q denotes the forward shift operator. Hence, A general linear model combining both parts could be written as:

$$y(k) = \frac{B(q)}{F(q)A(q)}u(k) + \frac{C(q)}{D(q)A(q)}v(k) \quad (2.1)$$

This general model operates as a unified framework for methods most commonly used in control engineering. Several models that belong to the classical approach in system identification have been established based on the previous general mathematical formula [5]. This includes the models: AR, ARMA, ARMAX, Box Jenkins and the nonlinear extensions NARX and NARMAX, which are considered to be the most applicable methods in a broad range of dynamic system problems. Nonlinear methods are extended in a straightforward fashion by replacing the simple linear polynomials with nonlinear equations [6-10].

Several restrictions apply on models based on mathematical forms. In practice, the type of nonlinearity is usually unknown, hence large number of coefficients stem from considering all forms of nonlinear equations under investigation. The models often run into difficulty if the system is noisy or features sparse measurements. Inherently nonlinear behaviour usually prevents the achievement of a reasonably valid model with the accurate structure. Therefore,

linear mathematical models are not the perfect representation due to potential shortcomings in accuracy over its full operation domain. Those shortcomings in the theoretical models increase the need of the development of more high-level techniques to handle nonlinear systems [11].

As a result of the growing complexity of nonlinear system behaviour, Artificial Intelligent techniques such as Soft Computing could be utilized as a suitable approach to model processes utilizing input/output experimental data. Soft computing consists of different ideas and procedures, which emphasize on building intellectual technologies to overcome the complications and imprecise conditions commonly encountered in real world problems. The ultimate target is to develop smart machines, which will interact in a similar way as human beings and integrate the clarity of thinking, and dealing with uncertainty with machine intelligence capability. L. A. Zadeh proposed and defined the term Soft Computing as a collection of methodologies that aim to exploit the tolerance for uncertainty and imprecision to achieve robustness, tractability, and low solution cost [12]. Soft computing trades the complex and slow techniques of hard computing with more intellectual handling techniques. The key stage for moving from hard to soft computing is the consideration that the compulsory computations struggle by conventional approaches.

Soft computing origins are associated to biological or interactive behavior in humans and include artificial neural networks (ANN) that comprehend intelligence through the imitation of neurons functionality in human's brain, fuzzy logic (FL) which realizes intelligence through the imitation of human reasoning treat, and Evolutionary Computation (EC) that comprehends intelligence through the imitation of genetic evolution. Neural networks are widely used for modelling and classification applications. Fuzzy logic delivers a natural framework for the processes via the concept of uncertainty. Genetic algorithms (GAs) belong to the EC family and are involved in various optimization and search processes. In this chapter, a brief outline of Fuzzy Logic and Neural Network systems is provided together with the hybrid approaches employing other soft computing methods.

2.4.2 Fuzzy Systems

Fuzzy systems include both fuzzy sets and fuzzy logic theory to deliver a significant accumulation to represent data of imprecise nature. In general, the concept of fuzziness allows dealing with ambiguity and vagueness usually traced in the human knowledge in a more conceptual framework [13]. The seminal paper proposed by Zadeh titled 'fuzzy sets' in 1965 [12] laid the foundation of fuzzy sets theory and fuzzy logic as a natural extension of the typical binary logic.

In classical set theory, the universe of discourse is strictly divided into two distinct crisp subsets, where an element either belongs to a given set or its complement. However, fuzzy sets theory abolishes this sharp dichotomy with gradual transition membership value that lies between 0 and 1. The difference between crisp and fuzzy sets is basically acknowledged by the membership function. In a similar manner, the fuzzy logic distinguishes propositions by allowing them to hold values that are not simply true T or false F. The degree of truth is based on membership function that operates over the range of real numbers $[0,1]$. This non-binary logic is a generalization of classic logic and proposes an excellent computational concept to capture the subjective human reasoning of the real processes.

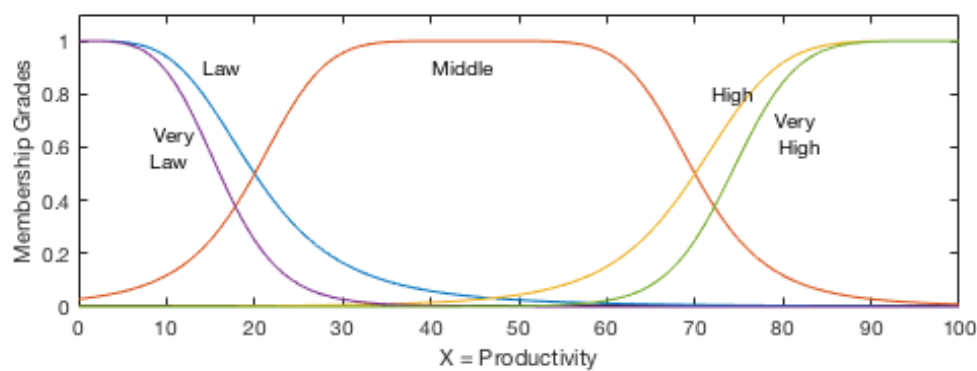


Fig. 2.3 “Productivity” membership functions

Variables, which take words instead of numerical values, are called linguistic variables, and they could be decomposed into a set of linguistic labels. For example, countless variables used in economics and finance such as credibility, profitability and reputation could suffer from ambiguity and subjectivity. While their approximations are highly desirable, the crisp logic fails to represent their uncertain nature, and as such this requires a more sophisticated tool to estimate them. For example, if we consider productivity as a linguistic variable, then using linguistic terms we could use expressions such as low, very low, middle, high and very high. Each one of these linguistic terms can be observed as a fuzzy subset that provides a basis for approximate reasoning. Fig. 2.3 illustrates the variable productivity expressed by its different linguistic terms using corresponding membership functions.

2.4.2.1 Identification with Fuzzy Modeling

The incorporation of fuzzy logic into data analysis provides human reasoning power to an otherwise simpler knowledge based scheme. The complexity in nonlinear problems can be dealt with using fuzzy logic as approximate reasoning forms the basis for its data

manipulation. Fuzzy inference systems conduct the approximate reasoning procedure, and consist of four main principles [14]:

- The fuzzification step evaluates crisp data according to membership functions in order to devise membership values for each linguistic variable. A number of membership functions are used in the literature, such as triangular, trapezoidal and Gaussian.
- A fuzzy IF-THEN rule is constituted of the premise and consequent parts, and is generally expressed as “If X is A then Y is B”, where A and B are linguistic values. The firing strength of each rule is obtained by using fuzzy operators, such as minimum and multiplication, on the values of the premise part.
- Approximate reasoning is performed to obtain a reasonable decision based on the membership values and the firing strength of each rule measured in previous steps.
- Lastly, the defuzzification step comprises of producing crisp output by using the aggregated consequents. This utilizes methods such as centroid of area, mean of maximum, or largest of maximum.

The main advantage gained of the fuzzy systems approach is the capability to express human knowledge and subjectivity using the IF-THEN rule structure with linguistic variables. In this sense, fuzzy reasoning is appropriate to be used in problems that need to understand human intuitive thinking. This understanding is inserted while building the model rather than undertaking it on the conclusion stage only. The natural rule representation as the basis of fuzzy logic ensures the model to be easy to interpret. This observation highlights many of the other fuzzy logic characteristics to allow the expert to specify the rules underlying the system behavior and represents the feature of each variable with fuzzy sets. In addition, new additional rules are easily extended without the need of rebuilding the knowledge base from scratch. One of the major drawbacks of fuzzy systems is their lack of learning ability to tune the fuzzy rules and membership parameters. Normally, experts would decide the fuzzy rules according to the available information and test it against the testing data. However, they would have to continuously adapt the system in a manual way and not to rely on the machine power to automatically adjust the rules and parameters. Fixed architecture is another rising problem in fuzzy systems. It is almost impossible to define the optimal features relying only on the human interaction, due to the uncertainty and complexity of the identifying system.

2.4.3 Neural Networks

Neural networks are information-processing paradigms that inspired by the biological neurons in human nervous system. They are designed to mimic a human brain approach to managing

knowledge. A neural network is composed of interconnected components (neurons) operating to perform a miniature function each and the network overall output is their weighted sum. The modeling approach is achieved by using the data to train a generic network to identify a suitable input/output relation approximation within a specific level of tolerance [15].

A neural network could be trained to learn the mapping that represents the required relation, by adjusting the values of the weights between the elements and accordingly modifying its internal structure. A key strength of NNs is the ability to generalise the trained knowledge and provide accurate responses to new data, making it a more adaptable and convenient tool of data modelling.

A neural network is formed according to a number of specifications. These include:

- The connectionist architecture between the neurons where signals travel
- The weight determination method of the parameters, or the learning algorithm
- The transfer functions of the neurons' inputs that determine the outputs.

Neural networks have a wide range of applications, including pattern recognition, function approximation, stock prediction, energy market prediction, image processing, weather forecasting, and security and loan applications [16-18]. Notably, NNs have outperformed conventional methodologies in solving problems of a complex nature. This is a result of NNs being trained to extract the needed information from the data and they work finest if the correlation between the inputs and outputs are highly non-linear. Neural networks aim to bring the conventional computers to mimic the human brain mechanism.

2.4.3.1 Architecture and learning in Neural Networks

A neural network commonly consists of an input layer, one or more hidden layers, and an output layer. The processing nodes of neural networks can be assembled in a range of layers depending on the desired purpose of the network. This arrangement of the neurons will form the overall structure of the neural network. Two significant architectures of neural networks are the feedforward and recurrent structures. Feedforward neural networks are characterised by the lack of loops in the structure, as the input layer is only mapped onto the output layer, but the reverse is not possible. Thus, network signals only move forward. Fig. 2.4 illustrates a renowned feedforward neural network known as the Multiple Layer Perceptron structure.

On the other hand, recurrent neural networks incorporate loops in their structure, which is the main component that distinguishes the two architectures apart. Fig. 2.5 illustrates an example of a recurrent neural network where the output is mapped back onto the input.

Network learning notation also known as algorithm training is the practice of modifying the weights and parameters to enforce a network to generate a particular result. Two different categories of training rules are distinguished

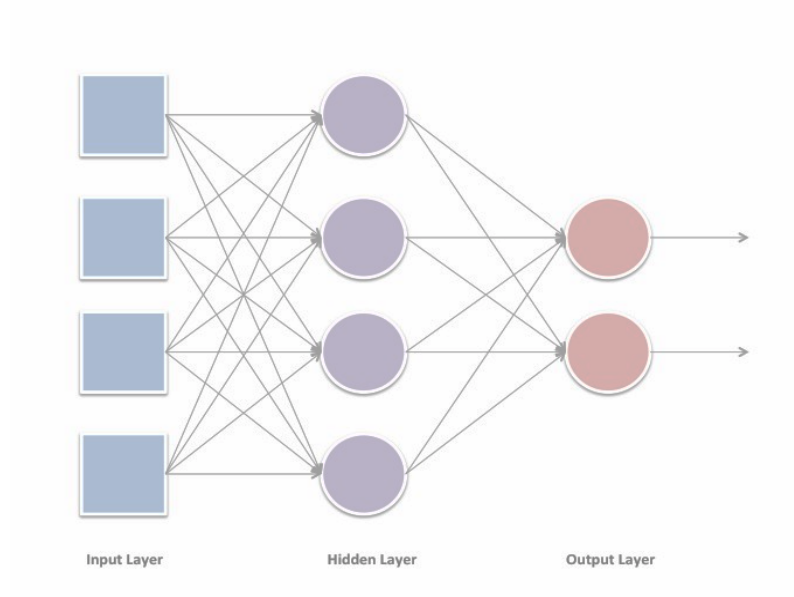


Fig. 2.4 Feedforward NN

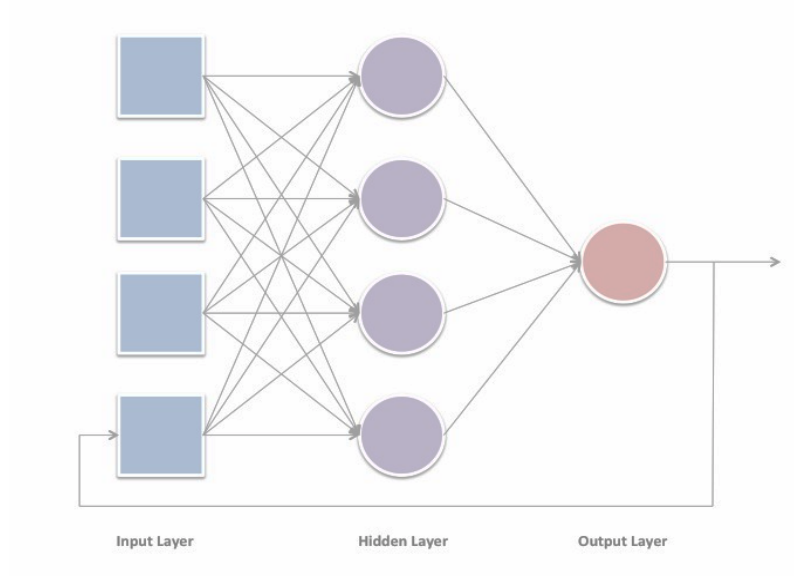


Fig. 2.5 Recurrent NN

I. Supervised learning of Neural Networks

The supervised rule is provided with training data that include a set of inputs and target outputs for the network to identify the system. As the inputs are applied to the model, the network result is compared to the system target. The rule of supervised learning is to adjust /tune the network parameters, which consistently can fulfill the input-output prerequisite for the entire system target. Among the typical rules that belong to supervised learning paradigm are delta rule and backpropagation training.

II. Unsupervised learning of Neural Networks

The desired output is not known in the unsupervised learning and hence there is no external signal to adjust the weights. In its place, the learning process internally monitors the algorithm performance until a balance condition is reached with stable weights. With the missing information, learning looks for trends and consistencies within the inputs signals and adapts the network function weights. Accordingly, the network organizes itself to build its topology and learn the parameters. The process of unsupervised learning is required in many clustering and pattern recognition problems. Hopfield rule [19] and Kohonen rule [20] are among popular types of unsupervised learning.

2.5 Hybrid Soft Computing Techniques

Intelligent hybrid systems using soft computing techniques have been identified as promising research field of computational intelligence (CI). The main hypothesis behind combining two or more soft computing algorithms, is to develop a hybrid technique that exploits the synergy between them, leveraging their benefits and overcoming their respective limitations [21]. This has indeed proven to be quite powerful for a variety of applications, such as pattern recognition, intelligent control, data mining and classification. Marked at the beginning of 1990s when scientists and researchers recognized that the hybrid process would lead to more powerful tools compared to solo methods. During the last few years there has been an energetic increase in research efforts aimed at synthesizing techniques. Smart combination of soft computing techniques is considered to be the new favorable frontline of Artificial Intelligence. Examples of promising hybridization paradigms that are widespread in practice are outlined in the following subsections.

2.5.1 Neurofuzzy Systems

A hybrid system known as a Neurofuzzy network combines the strengths of both Fuzzy Logic (FL) and Neural Networks. This integration allows a system to deal with data characterized with being both fuzzy and crisp, which is often the case in nonlinear problems. A brief comparison between fuzzy systems and neural networks from many aspects of knowledge is shown in Table 2.1. Fuzzy logic enables a model to approach complex problems that suffer from uncertainty effectively as it uses approximate reasoning. This means that more power is given to the user to determine a systems' behaviour through the FL basic principle, the IF-THEN rule, and the ability to input linguistic variables [22]. However, this could also be a downfall to the performance of a fuzzy logic based model if the number of input variables is large, which is known as the *curse of dimensionality*.

| Methodology | Fuzzy Systems | Neural Networks |
|------------------------------------|---|--|
| Presentation with natural language | Explicit implementation High flexibility | Implicit implementation Low flexibility |
| Learning and adapting | Low Induction | Highly skilled Adjusting parameters |
| Knowledge acquisition | Human expert interaction | Data sets |

Table 2.1 A comparison between Fuzzy Systems and Neural Networks

The linguistic aspect of FL also restricts its application to fields that lack the required linguistic information. NNs are advantageous as a complement to a neurofuzzy system due to their ability to adapt and modify their weights and behaviour to best fit the model. Generally, NNs strength is its ability for generalisations and computations. However, the learning process in MLP is not time effective and is often relatively slow. Moreover, the fundamental process in NNs known as the black box method is a downside as the networks actions is unknown and thus difficult to analyse.

2.5.2 Neural Genetic algorithms

Evolutionary Computing (EC) models are a class of stochastic search and optimization methods that are inspired by the Darwinian-type survival of the fittest strategy of creating offspring. By inheritance, the new descendants are expected to be fit individuals and significantly skilled to adapt the harsh environment. EC comprises Genetic Algorithms and Genetic Programming. Genetic algorithm (GA) is an optimum search-technique that mimics the practices of genetic evolution and natural selection to identify the fittest solution. The algorithm deals with a population of candidate solutions called individuals, which is usually represented by the shape of a string similar to genetic chromosome coding [23]. Usually, solutions are symbolised in binary values of 0s and 1s, although other encodings are available, they are less used. Unlike other optimization techniques, GA does not require mathematical descriptions of the optimization problem, but instead relies on a cost function in order to assess the fitness of a particular solution to the problem [24]. A classic genetic algorithm requires a genetic representation and a fitness function to assess the solution range. The algorithm improves the initialized population through applying an iterative set of stochastic operators (crossover and mutation) to the coded population in order to select fitter individuals.

Evolutionary artificial neural networks refer to a class of neural networks in which evolution optimization scheme is utilized to adapt the model. Because GA is virtuous at searching complex, non-differentiable and large-scale spaces, the GA fusion with NNs are explored to solve non-linear problems. GA methods are basically used for recognizing the topology and/or tuning the weights in neural networks. For instance, NNs using Backpropagation are faster to converge than GAs due to their local knowledge capability. However, this frequently comes with the risk of the NN getting stuck in local minima. Although GA techniques are not exposed to this dilemma, they are characterized by slow global search. For the drawbacks, GAs requires more functional computations compared to linearised techniques. Despite the great advantage of converging to a solution without exploiting derivatives, the method gives no assurance to find the function's minima [25].

2.5.3 Fuzzy Genetic Algorithms

The hybrid field of Fuzzy Genetic algorithms has been created through the combination of the two soft computing techniques. This is consistent with our previously debated conception about the ability of hybrid methods to exploit benefits and overcome the weaknesses of the original methods. One approach of hybridization is recognized when the FL controls the GAs resources. FL has the full ability to manage the GA instruments such as population size and weights of the altered selection operators. The genetic algorithm tools managed by FL result in an adaptive method that significantly improve efficiency and speed of convergence [26].

On the other hand, GA Fuzzy systems could get fine-tuned by the global searching talent of GAs. FL controller augmented by genetic algorithm is essentially a fuzzy inference system with genetic based learning process. The provision of a suitable fuzzy knowledge base (FKB) system is equivalent to optimizing the rules structure and membership function parameters with respect to the design conditions and constraints. Fuzzy logic based controllers is a prominent application of the FKB systems [27]. The knowledge base parameters are transformed into an appropriate genetic array where the evolutionary optimization process could perform. First the initial rule base is initialized using an intuitive investigation scheme, and then the algorithm would generate an improved rule base. Finally, GA will tune the membership parameters function in the antecedent part functions.

Chapter Three

COMPUTATIONAL INTELLIGENT SYSTEM PRINCIPLES

3.1 Fuzzy Systems

A fuzzy system estimates by inference an unknown mapping between two spaces from a set of logical statements. It incorporates a rule-based approach to solve a modelling problem rather than attempting to model a system mathematically. Fuzzy systems are based on fuzzy set and fuzzy logic theory, both introduced by Zadeh [12, 28]. A fuzzy system uses fuzzy set theory and its operations to solve a given problem and represents the imprecision found in real-world problems using IF-THEN rules, which are expressed in a comprehensible language.

A system in general is a combination of components that as a whole operate on a vector of input functions of time $x(t) \in R^n$ for each t to produce a vector of output functions of time $y(t) \in R^m$ for each t . A fuzzy system in particular is a system that uses fuzzy logic to operate upon the input $x(t)$ to produce the crisp output $y(t)$ [29]. A fuzzy system (also known as fuzzy logic system, fuzzy inference system or fuzzy-rule-based system) is constructed upon fuzzy logic theory, which considers logical variables with gradual truth-values between true and false. This character makes it suitable for uncertain or approximate reasoning, especially for systems with complex models that are difficult to derive. It is capable of mimicking cognitive thinking and provides a framework by which nonlinear models can be learnt and readily understood by humans. Fuzzy systems have been successfully applied to a wide variety of fields, such as data classification, automatic control, expert system, pattern recognition and robotics [30]-[34].

Fuzzy logic is a form of multi-valued logic that is employed to handle the concept of partial truth, where the truth-value may range between completely true and completely false. The truth-values of variables take any real number between 0 and 1, in contrast to the classical Boolean logic where the truth-values of variables may only be the integers 0 or 1. Furthermore, when linguistic variables are used, these degrees may be managed by specific membership functions. The concept of graded membership provides a mathematical precision to information arising from our cognitive process [29]. It is seen as a technique based on the

key notion that the activity of the human brain is not numbers but rather indicators of fuzzy sets where the transition is gradual between inclusion and non-inclusion. Having this main characteristic of fuzzy logic as an extension to classical logic means it is easier to deal with imprecise concepts in a well-defined way.

3.1.1 Fuzzy Sets and Membership Functions

Fuzzy set theory is a mathematical tool for translating abstract concepts found in natural language into computable entities. A classical set A is a set with a crisp boundary whereby an element either belongs to A or not. Although classical set theory is suitable for various applications and proved to be an essential part of mathematics, it does not reflect the imprecise nature of human concepts and opinions. In contrast to a classical set, a fuzzy set as the name indicates, is a set without a crisp boundary. The switch from 'belonging' to 'not belonging' to a set is gradual and is characterised by a transition function. This smooth move gives fuzzy sets the flexibility to model commonly used linguistic expressions. The theory of fuzzy sets provides a mechanism for representing linguistic values such as 'many', 'low', 'medium', 'often', and 'few'.

All elements x in the universe of discourse U are assumed to have a degree of membership through a characterized function μ called the **Membership Function** (MF) defined as:

$$\mu[x]: U \rightarrow [0,1]$$

where the MF maps each element of U to a membership grade or value between 0 and 1.

By definition, a **Fuzzy set** A in U is mathematically represented as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in U\} \quad (3.1)$$

where $\mu_A(x) = \mu[A(x)]$, represents the grade of membership of x belonging to the fuzzy set, and A is the fuzzy label or linguistic value (term) describing the linguistic variable. The universe of discourse might be a continuous space or consist of discrete elements.

Linguistic variables, which take text words rather than numerical values, could be decomposed into a set of linguistic labels or terms. Each label is expressed by a fuzzy set. In fuzzy theory the determination of fuzzy sets and MF relies on the knowledge of human experts [14].

Typical MFs for these linguistic values are displayed in Fig. 3.1 and defined by their corresponding equations (3.2)-(3.5). The universe of discourse X should be totally covered by the MFs and the transition from one MF to another should be smooth and gradual. The input space is divided into a number of regions, which can be labelled by a linguistic term. Each term is associated with a membership function that produces a membership value for each region.

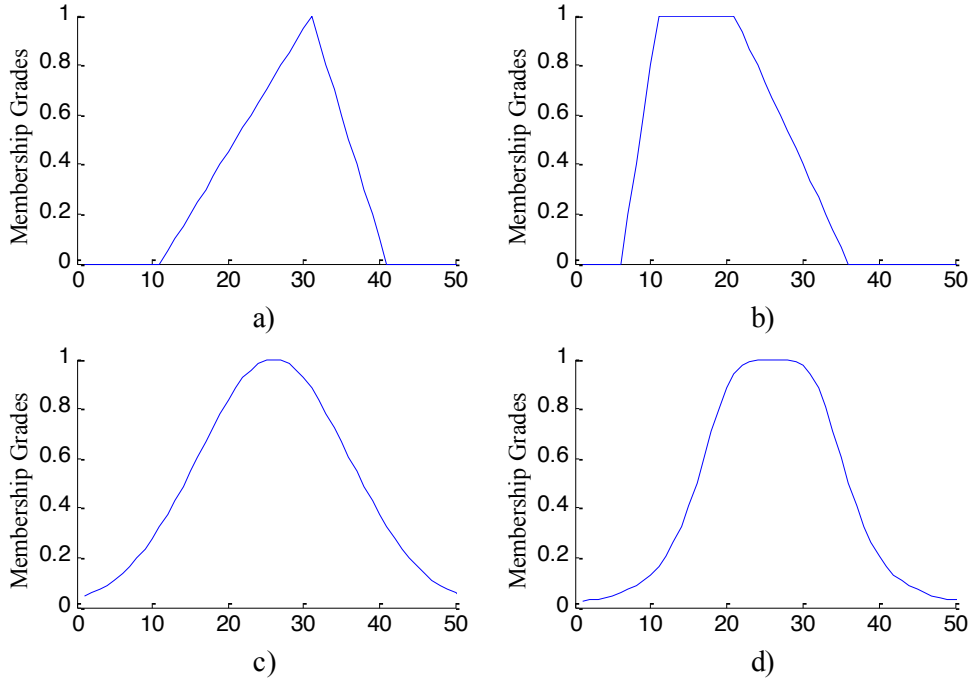


Fig. 3.1 Classes of membership functions:
a) Triangle, b) Trapezoid, c) Gaussian, d) Bell shaped

$$Triangle(x, a, b, c) = \max \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (3.2)$$

$$Trapezoid(x, a, b, c, d) = \max \left(\min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \quad (3.3)$$

$$Gaussian(x, c, \sigma) = e^{-\frac{1}{2} \left(\frac{x-c}{\sigma} \right)^2} \quad (3.4)$$

$$Bell_shaped(x, a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (3.5)$$

In general, the fuzzy set theory provides an inference structure that enables approximate human reasoning capabilities. On the contrary, the traditional binary set theory describes crisp events, events that either do or do not occur. It uses probability theory to explain if an event will occur, measuring the chance with which a given event is expected to occur.

The construction of a fuzzy set depends on two elements: the identification of a suitable universe of discourse and the specification of an appropriate membership function. The specification of a MF is subjective, which means that the MF specified for the same term by different experts may vary considerably. This subjectivity comes from individual perception and the expression of abstract concepts with no influence of randomness. Thus, subjectivity and randomness of fuzzy sets are the primary differences between fuzzy logic theory and probability theory, which deals with objective solutions of random phenomena [35].

3.1.2 Fuzzy Rules and Fuzzy Reasoning

The theory of fuzzy set is based upon the notion of relative graded membership and so are the functions of mentation and cognitive processes. Thus, the utility of fuzzy sets lies in their ability to model uncertain or ambiguous data often encountered in real life.

When more fuzzy sets are incorporated in a system, they can be connected using IF-THEN statements, forming a sequence of fuzzy rules. A form of a fuzzy IF-THEN rule (also known as fuzzy implication or fuzzy conditional statement) in a fuzzy system is given by

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B, \quad (3.6)$$

where A and B are fuzzy sets. The first part “ x is A ” of a rule evaluates the antecedent or premise, and involves fuzzification of the input. The second part “ y is B ” applies the result of the IF-part to the consequent or conclusion. The IF-part can have multiple antecedents, which are connected with fuzzy operators.

Three main operators which were defined by Zadeh [28] are fuzzy intersection “ \cap ” (AND), fuzzy union “ \cup ” (OR) and fuzzy complement “ $\bar{\cdot}$ ” (NOT), which are described in the following equations. It is assumed that A and B are fuzzy sets with MFs $\mu_A(x)$ and $\mu_B(x)$ respectively.

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x)) = \min(\mu_A(x), \mu_B(x)) \quad (3.7)$$

$$\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x)) = \max(\mu_A(x), \mu_B(x)) \quad (3.8)$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (3.9)$$

The class of fuzzy intersection operators is often referred to as T-norm and the fuzzy union class is known as T-conorm (S-norm). Some of the most frequently used T-norm and T-conorm (S-norm) operators are:

$$\text{Minimum} \quad T_{\min}(a, b) = \min(a, b) \quad (3.10)$$

$$\text{Algebraic product} \quad T_{ap}(a, b) = a * b \quad (3.11)$$

$$\text{Maximum} \quad S_{\max}(a, b) = \max(a, b) \quad (3.12)$$

$$\text{Algebraic sum} \quad S_{as}(a, b) = a + b - ab \quad (3.13)$$

Based on the interpretation of fuzzy intersection and fuzzy union, various T-norm and T-conorm operators are defined and used to formulate a number of qualified methods to calculate the fuzzy relations.

Fuzzy reasoning (also known as approximate reasoning) is the inference operations executed upon a set of fuzzy IF-THEN rules to derive conclusions. The steps of fuzzy reasoning performed by fuzzy inference systems are [22]:

1. **Degree of membership:** Compare the input variables with the membership functions on the premise part to obtain the membership values of each linguistic label. This step is called fuzzification.
2. **Firing forte:** Combine the membership values on the premise part to get firing strength of each rule through a specific T-norm operator, usually multiplication or min.
3. **Generation:** Apply the firing strength to the consequent MFs to produce the qualified consequent (either fuzzy or crisp) of each rule depending on the firing weight.
4. **Aggregation:** Aggregate the qualified consequents to produce an overall crisp output. This step is called defuzzification.

Fuzzy rules and fuzzy reasoning are the backbone when constructing the most important modeling tool based on fuzzy theory. Fuzzy Inference Systems (FIS) are built of Fuzzifier, a knowledge base called Fuzzy Inference Engine and Defuzzifier. A fuzzy inference engine combines a Fuzzy Rule Base into a mapping from the inputs of the system into its outputs, using fuzzy reasoning mechanisms. That is, fuzzy systems represent nonlinear mapping lead by fuzzy IF-THEN rules from the rule base, where each of these rules describes the local mappings [13].

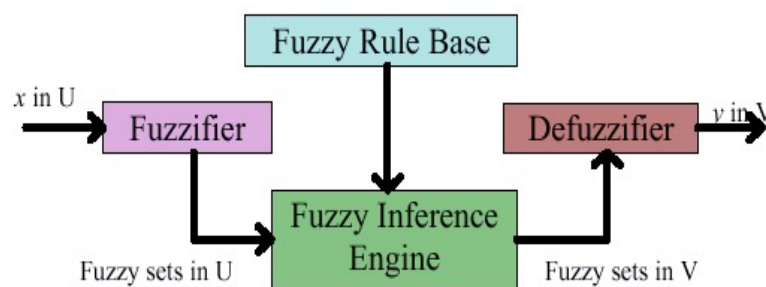


Fig. 3.2 Fuzzy Inference System structure

The fuzzy rule base is constructed either by a skilled expert or through applying automatic generation and extraction of rules using numerical input-output data. The units of a fuzzy inference system are shown in Fig. 3.2. After deriving the IF-part, the output of the THEN-part will be the result of a fuzzy reasoning process by applying the appropriate fuzzy operators. The output of an inference engine is an area with an irregular shape that depends on the membership functions used in the consequent part, the membership value of each rule and the inference type used.

The fuzzy IF-THEN for Multi-Input-Multi-Output (MIMO) rules are written in the following general form:

$$IF \ x_1 \text{ is } F_1, x_2 \text{ is } F_2, \dots, \text{ and } x_n \text{ is } F_n \ THEN \ z_1 \text{ is } C_1, z_2 \text{ is } C_2, \dots, \text{ and } z_p \text{ is } C_p \quad (3.14)$$

And the following form presents the fuzzy rules with Multi-Input-Single-Output (MISO):

$$IF x_1 \text{ is } F_1 \text{ and } \dots \text{ and } x_n \text{ is } F_n \text{ THEN } z \text{ is } C \quad (3.15)$$

where F_i is fuzzy set in space of discourse $U_i \in R$, C is fuzzy set in $V \in R$; and x_i and z are measured variables. Most researchers use and analyse only Multi-Input-Single-Output (MISO), since a Multi-Input Multi-Output (MIMO) can be decomposed into a multi-input single output for each output. It can be shown that a fuzzy system with n inputs and m outputs (MIMO) is equivalent to m fuzzy systems, each with n inputs and one output (MISO) fuzzy systems.

3.1.3 Defuzzification Methods

Defuzzification is the process of transferring a fuzzy output into a crisp output using some chosen defuzzification method. This method extracts a crisp value that represents a fuzzy set. Two types of fuzzy inference systems that have been widely used in various applications are the Mamdani model [36] and the Takagi, Sugeno and Kang (TSK) model [37, 38]. The distinction between them lies in the consequent part of the fuzzy rules where different aggregation and defuzzification procedures are employed.

3.1.3.1 The Mamdani inference model

The Mamdani system was proposed as the first attempt to control a steam engine and boiler combination by a set of linguistic rules obtained by experienced human operators. In the Mamdani method consequents are fuzzy sets, and the final output is based on defuzzification of the overall output.

Fig. 3.3 is an illustration of a two-rule Mamdani inference system employing the min and max aimed at T-norm and T-conorm operators, respectively.

The form of the rules is

$$\begin{aligned} &IF \ x \text{ is } A_1 \text{ AND } y \text{ is } B_1 \text{ THEN } z \text{ is } C_1 \\ \text{Also } &IF \ x \text{ is } A_2 \text{ AND } y \text{ is } B_2 \text{ THEN } z \text{ is } C_2 \end{aligned}$$

The Mamdani method has several variations [21]. There are different t-norms to use for the connectives of the antecedents, different aggregation operators for the rules, and numerous defuzzification methods that could be used.

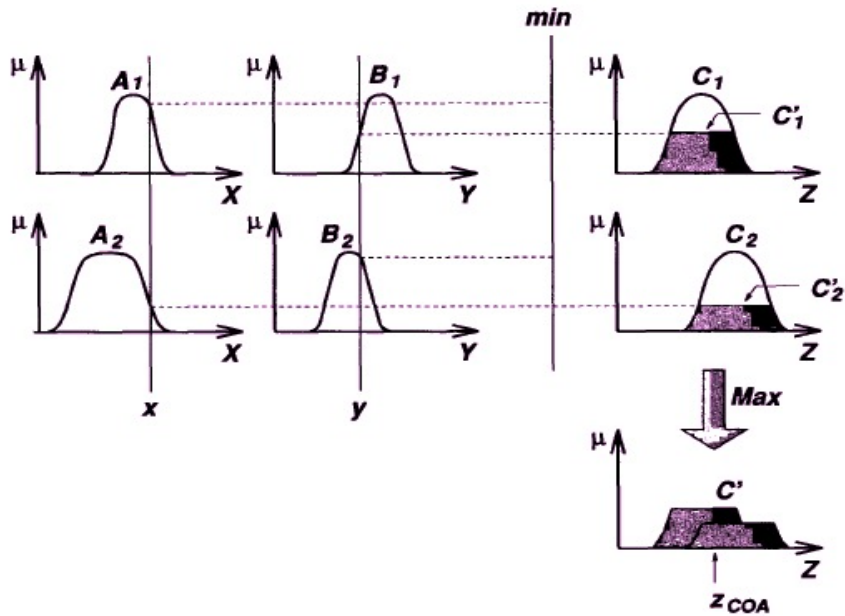


Fig. 3.3 The original Mamdani FIS using (max-min) composition

One alternative composition is to use product for the implication operator and summation for the aggregation operator. This **sum-product** variation has the advantage of a simpler defuzzification scheme with computation shortcut.

Some of the mostly used defuzzification methods are centroid of area, centre average, bisector of area, mean of maximum, smallest of maximum, or largest of maximum (Fig. 3.4).

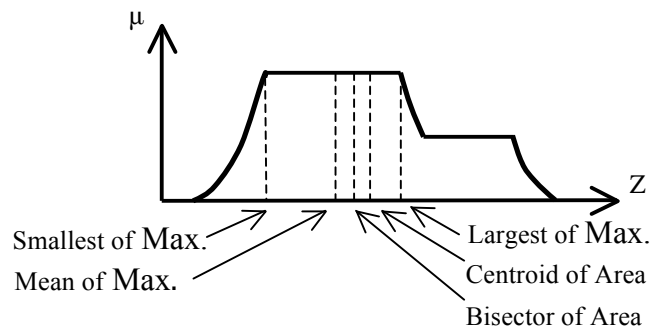


Fig. 3.4 Various defuzzification methods

The Centroid approach (also called center of area or center of gravity) is a logical answer to defuzzification because it uses all available information to compute the output. Centroid defuzzification can be put into equation form as

$$z_{centroid} = \frac{\int \mu_A(z)z dz}{\int \mu_A(z) dz} \quad (3.16)$$

where $\mu_A(z)$ is the aggregated output MF. Centroid is one of the most widely adopted techniques, however it suffers from the intensive computation dilemma.

Under the **sum-product** composition, the output of a Mamdani model with centroid defuzzification method is equal to the weighted average of the centroids of the consequents MFs [22]. The method is called centre average (CA) defuzzifier and has the formula

$$z_{CA} = \frac{\sum \mu_A(z)z}{\sum \mu_A(z)} \quad (3.17)$$

where each of the weighting factors is equal to the product of a firing strength and the consequent MF's area. The problem with the CA defuzzifier is that it suffers due to not using the entire shape of the consequent membership function

3.1.3.2 TSK inference model

The second inference method, generally referred to as the TSK method, or alternatively Sugeno method was proposed in an effort to develop a systematic approach to generating fuzzy rules from a given input–output data set [37]. TSK fuzzy model is one of the most outstanding fuzzy models in the literature, which is suitable to model a large class of non-linear systems.

A typical rule in a TSK model, which has two inputs x and y and one output z , has the form

$$IF \ x \text{ is } A \text{ and } y \text{ is } B \ THEN \ z = f(x, y) \quad (3.18)$$

where $f(x, y)$ is a crisp function in the consequent part. Usually $f(x, y)$ is a polynomial function in the inputs x and y , but it can be any general function as long as it describes the output of the system within the fuzzy region specified in the antecedent of the rule to which it is applied. When $f(x, y)$ is a constant, the inference system is called a zero-order Sugeno model, which is a special case of the Mamdani system in which each rule's consequent is specified as a fuzzy singleton. When $f(x, y)$ is a linear function of x and y , the inference system is called a first-order Sugeno model.

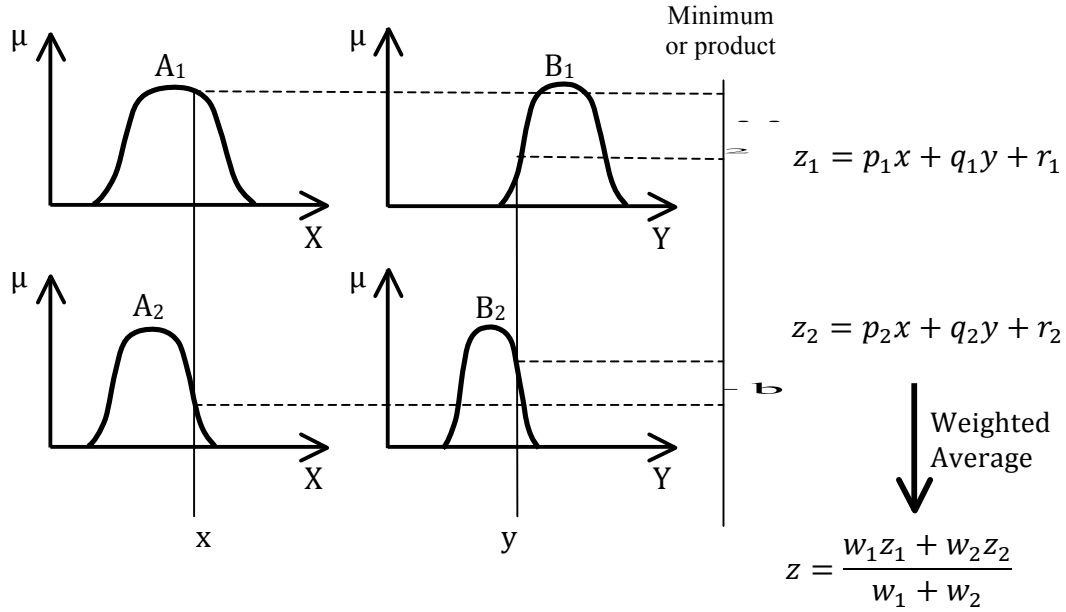


Fig. 3.5 TSK fuzzy model

Fig. 3.5 illustrates two rules TSK model where each rule has a crisp function in the consequent part with a linear polynomial defined as follows:

$$\text{IF } x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z = px + qy + r \quad (3.19)$$

The overall output is obtained via weighted average, thus avoiding time-consuming defuzzification that is required in the Mamdani model.

TSK fuzzy model consists of a number of local linear models, possessing excellent ability to describe uncertain systems and to approximate a nonlinear model with any given accuracy. The basic idea of this method is to decompose the input space into “fuzzy partitions” and to approximate the system in every region by a simple piecewise linear model. The overall fuzzy model is thus considered as a combination of interconnected subsystems with simpler models. Typically, in a general TSK model, the employed IF–THEN rules can be viewed as the expansion of piecewise linear partition [39] and they are presented as

$$\begin{aligned} \text{Rule}_i: & \text{ IF } x_1 \text{ is } F_{i,1} \text{ and } x_2 \text{ is } F_{i,2} \text{ and } \dots x_M \text{ is } F_{i,M} \text{ THEN} \\ & z_i = a_{i0} + a_{i1}x_1 + \dots + a_{iM}x_M, \text{ where } i = 1, 2, \dots, N \end{aligned} \quad (3.20)$$

The Rule_i represents the i^{th} fuzzy inference rule, $x = (x_1, x_2 \dots x_M)$ is the premise fuzzy variables and $F_{i,M}$ are the fuzzy sets. The consequent parts of the rules are specified using polynomials, which are linear equations with parameters a_{ij} belonging to i^{th} rule and j^{th} input variable where $(i=1, \dots, N)$ and $(j=1, \dots, M)$. The functioning range of any fuzzy rule is defined by the membership functions of the antecedent part.

The overall output z of the N-rules in TSK fuzzy system is aggregated as a weighted sum of fuzzy rule outputs. The defuzzification method applied is given by:

$$z = \sum_{i=1}^N (a_{i0} + a_{i1}x_1 + \dots + a_{iM}x_M) * \mathcal{F}_i(x) \quad (3.21)$$

Where $\mathcal{F}_i(x)$ is the normalised firing strengths of the rule i and obtained as

$$\mathcal{F}_i(x) = \frac{\prod_{j=1}^M \mu_i(x_j)}{\sum_{i=1}^N \prod_{j=1}^M \mu_i(x_j)}, \quad x = (x_1, x_2, \dots, x_M) \quad (3.22)$$

With $\mu_i(x_j)$ is the Gaussian membership of the fuzzy sets $F_{i,j}$ defined as

$$\mu_i(x_j) = \text{gaussian}(x_j; c_{ij}, \sigma_{ij}) = \exp \left(-\frac{1}{2} \left(\frac{x_j - c_{ij}}{\sigma_{ij}} \right)^2 \right) \quad (3.23)$$

$$\text{For } i = 1, \dots, N \text{ and } j = 1, \dots, M$$

where c_{ij} denotes the centres and σ_{ij} depicts the standard deviation for membership functions associated with rule i and input j . The parameters are obtained by fitting the equation (3.21) to the set of data points by numerical optimization methods.

The fundamental distinction between the two inference systems lies in the approach used to generate the crisp output from the fuzzy inputs. Mamdani uses an output membership function and obtain the crisp output through the defuzzification of the consequent rules. On the other hand, TSK method has no output membership function and crisp result is found using the weighted average of the rules. Mamdani type is recognised for describing the expert knowledge in more intuitive manner. This interpretability power is lost in TSK type since the consequent part of the rule base is not fuzzy. However, Mamdani type entails extensive computational load while TSK replaced the time-consuming defuzzification phase by the weighted average process. Moreover, TSK inference system support only the MISO type whilst Mamdani is capable of adapting both MISO and MIMO styles.

3.2 Neural Network Systems

Neural networks are biologically inspired computing structures that are conceptually modelled after the nervous system. In particular, they were created in an attempt to imitate the processing patterns of the human brain. The fundamental aspect of this paradigm is the novel design for processing information [15]. Although humans are not as fast or precise as digital computers, they are typically much better at perceiving cognitive tasks such as identifying an object of interest in a natural scene, or interpreting natural language, than a digital computer. The basic processing unit in the brain is the neuron (nerve cell), which is not individually an intelligent element. However, the hundred billion of interconnected neurons coupled with approximately three orders of magnitude more connections synapses are capable of understanding and representing knowledge [40]. A synapse is a structure that permits a

neuron to pass an electrical or chemical signal to another neuron. A neural network of processing artificial neurons is designed based on the physiology and individual processing neurons of the brain. Initial work by McCulloch and Pitts in 1943 presented simplified artificial neurons that were shown to have basic logical properties. In 1957 Frank Rosenblatt put forward the concept of the Perceptron and Widrow developed the first training algorithm called Adaline [16, 41].

The neural network incorporates highly connected processor units, which are mimicking organic neurons. The operational features of the total network are determined by the weight of each connecting link. Modifying the links' weight in an adaptive manner forms the basis of network learning. Neural computing is a study of networks of adaptable nodes, which, through a process of learning from task examples, store experiential knowledge and make it available for use. Once trained using the existing data, neural networks can subsequently recognize such patterns when they occur again. Moreover, to a suitable degree of accuracy, similar patterns are also detected by what is known as generalization.

Perceptron – Feed Forward

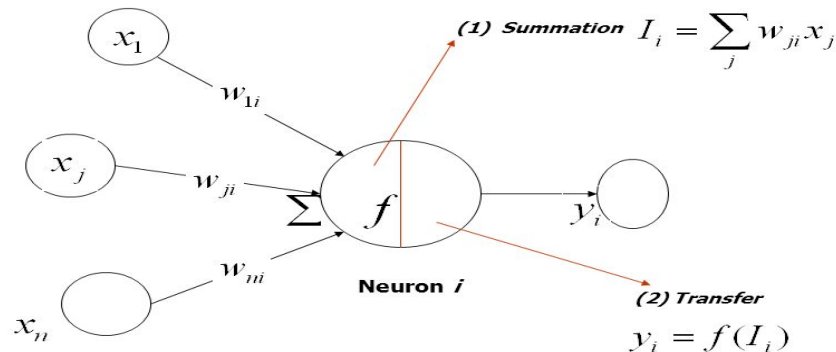


Fig. 3.6 Single perceptron process

Fig. 3.6 depicts the feed forward information process of a single neuron with n inputs, which are individually weighted. This model is based on the concept of the perceptron originated by Frank Rosenblatt in 1957 [41]. The input vector $x = (x_1, x_2, \dots, x_n)$ is multiplied by its corresponding weight $w = (w_1, \dots, w_n)$. The neuron includes a summation operator to combine the incoming information. The result is then passed through an arbitrarily selected activation (transfer) function.

Equation 3.24 presents the computed result for this simple neuron.

$$y_i = f(I_i) = f(\sum_{j=1}^n w_{ji} x_j) \quad (3.24)$$

The collective structure and learning ability empower the network to solve difficult and tedious problems. This is also coupled with the potential property of faster operational speeds achieved through underlying parallel operations. NNs are nonlinear systems due to the nonlinear activity of their unit's neurons. In several cases a nonlinear system is viewed as being sinister; however, this nature is highly appreciated, especially when NN is applied to model a training data acquired from nonlinear real processes.

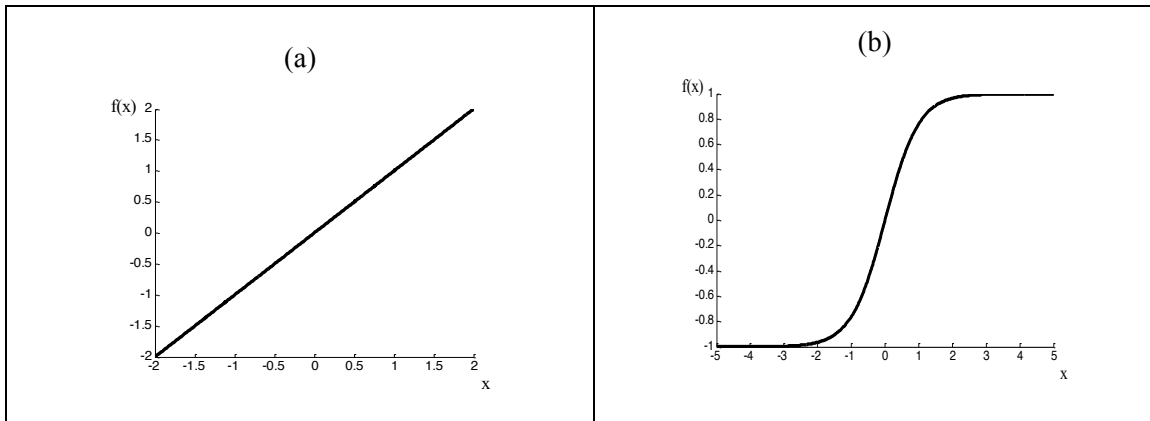
Neural networks follow the black box modelling style where no prior knowledge is a prerequisite to identify the required system. This modelling fashion is extremely flexible to model different mappings through learning from the input-output data. On the other hand, the stored knowledge in the output model is difficult to explain and interpret.

The field of neural networks has a strong interdisciplinary foundation, involving computing, biology, electronics engineering and neuropsychology. This attracts the interest of researchers in many different backgrounds to employ the NN in applications such as nonlinear system modelling, function approximation, prediction, clustering, data compression, pattern classification, feature extraction and nonlinear control [42-46]. The successful implantations have arisen partly due to the huge power of recent computers and the computing approach delivered by this paradigm to provide tractable solutions to complex problems.

3.2.1 Activation Functions

Some of the common activation functions found in NNs literature are:

- i. Linear function which transfers the summation output to the same value. [Equation 3.25, Fig. 3.7 (a)]



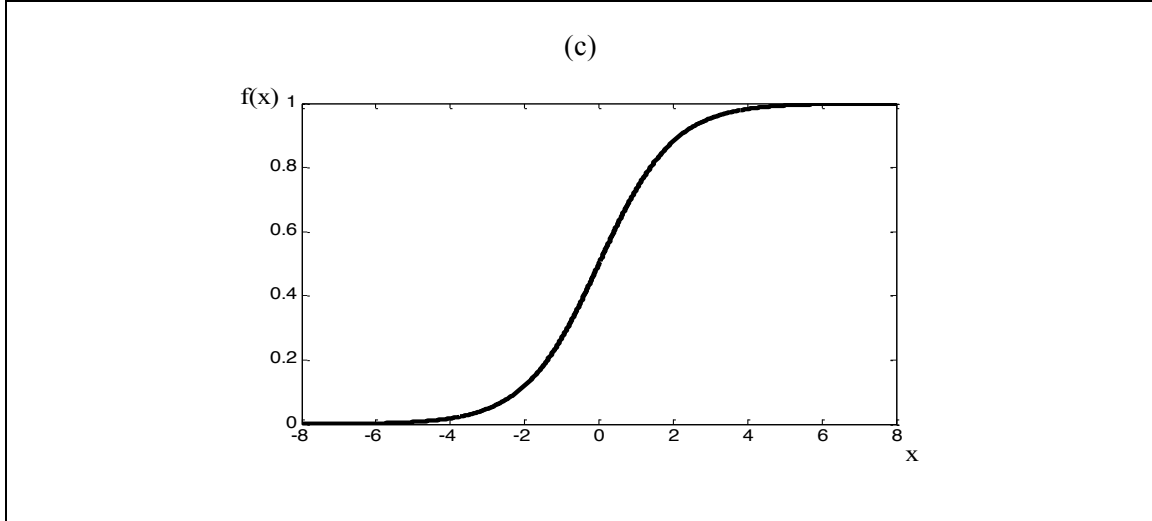


Fig. 3.7 Neuron Activation Functions
a) Linear, b) Hyperbolic tangent, c) Sigmoid

$$f(x) = \text{Purelin}(x) = x \quad (3.25)$$

$$f(x) = \text{Tangent}(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (3.26)$$

$$f(x) = \text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (3.27)$$

- ii. The differentiable nonlinear transfer functions hyperbolic tangent and sigmoidal function take $R=(-\infty, \infty)$ as their input domain, while their output range is $(-1,1)$ and $(0,1)$, respectively. [Equation 3.26, Fig. 3.7 (b) and Equation 3.2, Fig. 3.7 (c)].

3.2.2 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) networks are an important class of neural networks and have become the most encountered NN. Typically, the feedforward network consists of a set of source neurons (nodes) that represent the input layer, one or more intermediate hidden layers, and an output layer. Fig. 3.8 depicts the architecture of a MLP with two hidden layers and an output layer. The network shown is entirely connected, which means that a neuron in any layer must be linked to all the neurons in the former layer [16].

Activation functions are employed for both the hidden and output computation nodes where the output from one layer becomes the input to the next. Typically, the sigmoid function is used to transfer the output of the neuron from 0.0 to 1.0.

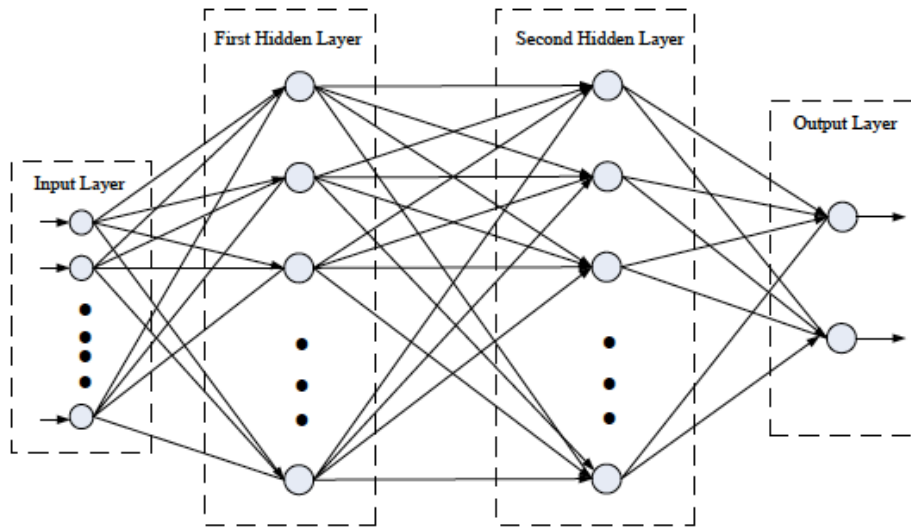


Fig. 3.8 MLP structure

The feedforward process starts by calculating the output of the first hidden layer using the input nodes values and the weights connecting both the input and the hidden layer, then calculates the next layer with the same fashion. The outputs of any hidden layer could be directed to multiple other neurons for further processing. As the learning process advances layer by layer across the MLP, the hidden nodes begin to gradually discover the outstanding features of the training data. The parallel nature of NN comes into performance with multiple neurons in a layer. Each of these neurons can process their functions at the same time, making neural networks faster in the MLP architectures.

Feedforward networks are generally static. They lack the memory units and their response to input signal is independent of the current network status. Conversely, Recurrent networks have a dynamic nature where the memory units nourish the network with feedback values once computed which leads the network to reach a further state [41].

One of the MLP shortcomings is the difficulty to explain the theoretical analysis of the network due to the form of distributed nonlinearity and the excessive connectivity between its elements. Furthermore, the implicit functionality handled in hidden layers makes the entire process harder to visualize, hence falling into the black box category.

3.2.3 Backpropagation Algorithm

In the paradigm of learning procedures, the Backpropagation algorithm is perhaps the most prominent method for the determination of the associated weights between the neurons. The algorithm essentially uses a training sample to enhance the fit of the network parameters by

comparing the partial derivative of the mean squared error between the system output and the desired output [47]. Given a training set of input/output data in the neural network layers, the error signal acquired from the output of each neuron is propagated backward to modify the previous layer's weights. The generalised delta rule is applied for adjusting the parameters of the feedforward networks in order to minimise a predetermined cost error function.

The mathematical basis for the backpropagation algorithm is the optimisation technique known as gradient descent method. The gradient of a function gives the direction in which the function increases most rapidly; a negative value of the gradient gives the direction in which the function decreases most rapidly.

Two reversal passes of calculations are distinguished in the application of the learning scheme. The *forward pass* is where the parameters stay unchanged throughout the network, and the function signals of the network are moving forward layer-by-layer. The second pass is the *backward pass*, which starts by recursively transferring the error signal from the output layer to compute the local gradient vector in every hidden layer [15, 16].

First we define some necessary notations to describe the algorithm mathematical equations.

- x_j^l represents the input to node j of layer l .
- w_{ij}^l is the weight from node i of layer $(l-1)$ to node j of layer l .
- y_j^l represents the output of node j of layer l
- d_j represents the desired output of the training data.
- $A(x)$ is an activation function where the sigmoid function is used in this description.

We can outline the backpropagation-learning algorithm as Gradient Descent on sum-squared error. The total error E in a network is given by the following equation

$$E = \frac{1}{2} \sum_{j \in L} (d_j - y_j^L)^2 \quad (3.28)$$

L is the total number of layers, and y_j^L is the output of the neural network corresponding to the input x . We want to adjust the network's weights to reduce this overall error. The rate of

change of the error with respect to the weights is giving by the delta: $\Delta w = -\eta \frac{\partial E}{\partial w}$, where η

is the learning rate, and the negative sign indicates that weight changes are in the direction of decrease in error. However the error in equation (3.28) is not a direct function of the weight.

Hence the chain rule is applied. Two placement of nodes are considered, an output node or a node belonging to a hidden layer

First, in the output layer we compute the partial derivative based on the chain rule. That is:

$$\begin{aligned}\frac{\partial E}{\partial w_{jk}^L} &= \frac{\partial}{\partial w_{jk}^L} \frac{1}{2} \sum_{j \in L} (d_j - y_j^L)^2 \\ &= (d_k - y_k^L) \frac{\partial}{\partial w_{jk}^L} y_k^L\end{aligned}\quad (3.29)$$

y_k^L is the output of activation function. So, the equation becomes:

$$\frac{\partial E}{\partial w_{jk}^L} = (d_k - y_k^L) \frac{\partial}{\partial w_{jk}^L} A(x_k^L) \quad (3.30)$$

where x_k^L is the linear combination of all inputs of the node j in the layer L with the weights.

The derivative of sigmoid function has the special form, which is easy to implement:

$$\begin{aligned}\frac{d}{dx} A(x) &= \frac{d}{dx} \left(\frac{1}{1 + e^{-x}} \right) \\ &= \frac{e^{-x}}{(1 + e^{-x})^2} \\ &= \frac{1 + e^{-x}}{(1 + e^{-x})^2} - \left(\frac{1}{1 + e^{-x}} \right)^2 \\ &= A(x) - A(x)^2\end{aligned}\quad (3.31)$$

Therefore, the partial derivative function becomes:

$$\frac{\partial E}{\partial w_{jk}^L} = (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L)) \frac{\partial}{\partial w_{jk}^L} x_k^L \quad (3.32)$$

The last term is based on chain rule, where $x_k^L = \sum_{i \in L-1} w_{ik}^L y_i^{L-1}$. Thus

$$\frac{\partial E}{\partial w_{jk}^L} = (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L)) y_j^{L-1} \quad (3.33)$$

By this equation, we find the relation between j node of $L-1$ layer and the k node of L layer.

We define the new notation $\delta_k = (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L))$

to represent the k node of the L layer term. So equation (3.33) becomes:

$$\frac{\partial E}{\partial w_{jk}^L} = \delta_k y_j^{L-1} \quad (3.34)$$

Secondly, consider the nodes in the l hidden layer. Similarly, we need to apply partial derivative over hidden layer weights on the error function

$$\begin{aligned}\frac{\partial E}{\partial w_{ij}^l} &= \frac{\partial}{\partial w_{ij}^l} \frac{1}{2} \sum_{k \in L} (d_k - y_k^L)^2 \\ &= \sum_{k \in L} (d_k - y_k^L) \frac{\partial}{\partial w_{ij}^l} y_k^L\end{aligned}\quad (3.35)$$

Again, we apply chain rule to get:

$$\frac{\partial E}{\partial w_{ij}^l} = \sum_{k \in L} (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L)) \frac{\partial}{\partial w_{ij}^l} x_k^L \quad (3.36)$$

Then, we modify the last derivative term by chain rule:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^l} &= \sum_{k \in L} (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L)) \frac{\partial x_k^L}{\partial y_j^l} \frac{\partial y_j^l}{\partial w_{ij}^l} \\ &= \sum_{k \in L} (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L)) w_{jk} \frac{\partial y_j^l}{\partial w_{ij}^l} \end{aligned} \quad (3.37)$$

Now, we simplify the derivative term based on the chain rule:

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^l} &= \sum_{k \in L} (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L)) w_{jk} A(x_j^l) (1 - A(x_j^l)) \frac{\partial x_j^l}{\partial w_{ij}^l} \\ &= \sum_{k \in L} (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L)) w_{jk} A(x_j^l) (1 - A(x_j^l)) y_i^{l-1} \\ &= A(x_j^l) (1 - A(x_j^l)) y_i^{l-1} \sum_{k \in L} \delta_k w_{jk} \end{aligned} \quad (3.38)$$

Define $\delta_j = A(x_j^l) (1 - A(x_j^l)) \sum_{k \in L} \delta_k w_{jk}$, therefore the equation becomes

$$\frac{\partial E}{\partial w_{ij}^l} = \delta_j y_i^{l-1} \quad (3.39)$$

Using all mathematical derivations, BP algorithm can be summarised in the following steps:

The Backpropagation Algorithm

1. Run the network forward with input data to get the network output
2. For each output node, compute

$$\delta_k = (d_k - y_k^L) A(x_k^L) (1 - A(x_k^L))$$

3. For each hidden node, compute

$$\delta_j = A(x_j^l) (1 - A(x_j^l)) \sum_{k \in L} \delta_k w_{jk}$$

4. Update the weights as follows:

Calculate

$$\Delta w = -\eta \delta_l y_{l-1}$$

Apply Gradient Decent method

$$w_{new} \leftarrow w_{old} + \Delta w$$

3.2.4 Wavelet Neural network (WNN)

In recent years, wavelets have become a very active subject in many scientific and engineering research areas [49]. Especially, wavelet neural networks (WNN) inspired by both the feed-forward neural networks and wavelet decompositions have received considerable attention and become a popular tool for function approximation [50]. The main characteristic of WNN is that some kinds of wavelet functions are used as the nonlinear transformation function in the hidden layer, instead of the usual sigmoid function. Incorporating the time-frequency localisation properties of wavelets and the learning abilities of general neural network, WNN has shown its advantages over the regular methods such as NN for complex nonlinear system modelling [51].

The fundamental of WNN is looking for a series of appropriate wavelet basis functions in wavelet space. This process can be realized by iteration computation of wavelet basis function, which is to make the energy function minimised. To design a WNN, the main work is to design network structure, to determine the number of nodes in the hidden layer, and to choose the wavelet basis function and the learning algorithm of weights training. A simple WNN is shown in Fig. 3.9 (a).

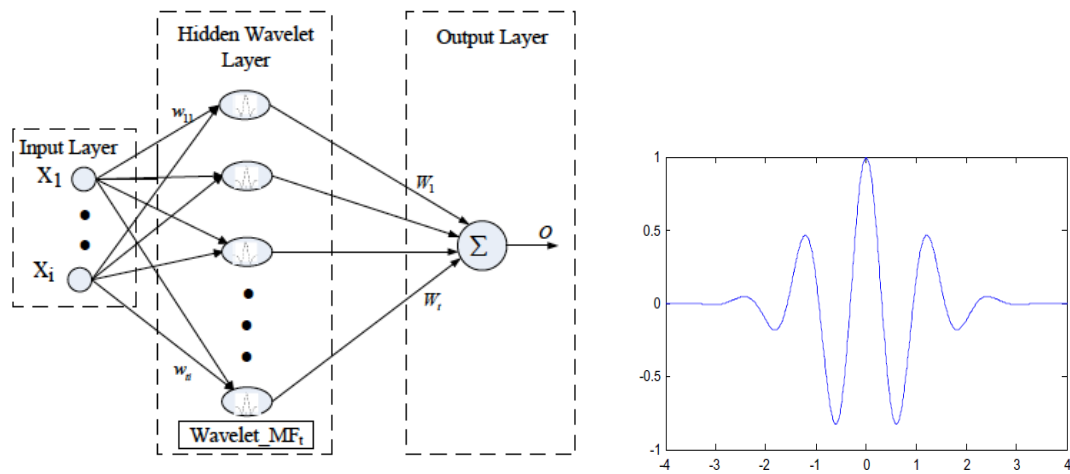


Fig. 3.9 (a) WNN structure (b) Morlet Wavelet basis function

The WNN consists of three layers: input layer, hidden layer and output layer. The connections between input units and hidden units, and between hidden units and output units are called weights w_{ij} and W_i respectively. In this WNN, the training procedure is described as follows:

- *Initialising the dilation parameter a_i , translation parameter b_i and node connection weights w_{ij} , W_i to some random values. All those random values are limited in the interval $(0, 1)$.*

- Input data $X_n(i)$ and the corresponding output values V_n , where i varies from 1 to S , representing the number of the input nodes, n represents the n^{th} data sample of training set, and V_n^T represents the target output state.
- The output value of the sample V_n is calculated with the following formula:

$$V_n = \sum_{t=1}^T W_t \psi \left(\frac{\sum_{i=1}^S w_{ti} x_n(i) - b_t}{a_t} \right) \quad (3.40)$$

Where ψ is considered a mother wavelet [49]. The Morlet wavelet filter is employed in this derivation and its plot is shown in Fig. 3-9(b). Morlet filter is represented by

$$\psi(t) = \cos(\omega_0 t) \exp(-0.5t^2) \quad (3.41)$$

- To reduce the error, W_t, w_{ti}, a_t, b_t are adjusted using $\Delta W_t, \Delta w_{ti}, \Delta a_t, \Delta b_t$. In the WNN, the gradient descend algorithm is employed, through the following equations,

$$\Delta W_t(j+1) = -\eta \frac{\partial E}{\partial W_t(j)} + \alpha \Delta W_t(j) \quad (3.42)$$

$$\Delta w_{ti}(j+1) = -\eta \frac{\partial E}{\partial w_{ti}(j)} + \alpha \Delta w_{ti}(j) \quad (3.43)$$

$$\Delta a_t(j+1) = -\eta \frac{\partial E}{\partial a_t(j)} + \alpha \Delta a_t(j) \quad (3.44)$$

$$\Delta b_t(j+1) = -\eta \frac{\partial E}{\partial b_t(j)} + \alpha \Delta b_t(j) \quad (3.45)$$

where the error function E is taken as:

$$E = \frac{1}{2} \sum_{n=1}^N (V_n^T - V_n)^2 \quad (3.46)$$

and N standing for the data number of training set, η and α being the learning rate and the momentum term, respectively.

- The process is continued until E satisfies the given error criteria, and the whole training of the WNN is completed.

3.3 Neurofuzzy Systems

The motivating force behind the combination of two or more of the soft computational techniques is to avoid limitations and emphasize on benefits in their hybrid creation. Following the study of the two soft computing methods: Fuzzy logic systems and NN, one

could point out both the advantages and disadvantages for each method. In pattern recognition for example, fuzzy logic can reason vague data and clearly explain the acquired result. However, they lack the automated ability to update the rules needed for their decision making process [52].

The problem of finding membership functions and appropriate rules is frequently a tiring process of attempt and error. This leads to the idea of employing learning algorithms to the fuzzy systems. The neural networks, that have efficient learning algorithms, had been presented as an alternative to automate or to support the development of tuning fuzzy systems. As a result, the performance of the hybrid model is significantly improved. The first studies of the neurofuzzy systems started on the 1991 with Jang [22], Lin and Lee [13], followed by Berenji [54] and Nauck [55].

The majority of the first applications were in process control. Gradually, its application spread for all the areas of the knowledge like, data analysis, data classification, imperfections detection and decision-making support systems [53, 56, 57].

3.3.1 Adaptive Neuro-Fuzzy Inference System (ANFIS)

One of the most frequently used neurofuzzy systems is the adaptive neurofuzzy inference system (ANFIS). The structure was proposed by Jang [22, 58] and it is based on the TSK reasoning configuration. In the TSK model, the output of each rule is a linear combination of the input variables. Fig. 3.10 demonstrates the ANFIS structure with two inputs and two IF-THEN rules of TSK system type:

$$\mathfrak{R}_1: \text{ if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } f_1 = p_1x_1 + q_1x_2 + r_1$$

$$\mathfrak{R}_2: \text{ if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } f_2 = p_2x_1 + q_2x_2 + r_2$$

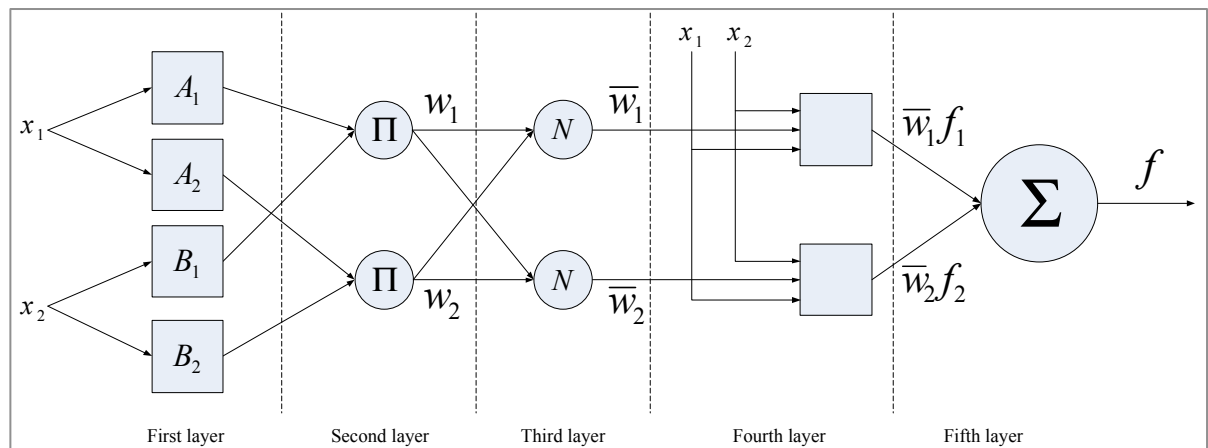


Fig. 3.10 A two-input ANFIS architecture

ANFIS Structure

A description of each layer as well as the learning process is discussed below layer by layer.

Layer 1: The output of the i^{th} node of this layer is

$$o_i^1 = A_i(x) \quad (3.47)$$

where x is the input to the i^{th} node and the linguistic label is associated with the Gaussian membership function A_i :

$$A_i(x) = \exp\left(-\left(\frac{x-c_i}{a_i}\right)^{2b_i}\right) \quad (3.48)$$

where $\{a_i, b_i, c_i\}$ is the parameter set of the premise part.

Layer 2: Nodes of this layer multiply the incoming signals and send the product out.

Each node output represents the firing strength w_i of the corresponding rule.

Layer 3: The outputs of the nodes in this layer are normalized firing strengths \bar{w}_i .

Layer 4: The output of the i^{th} node of this layer is based on the TSK defuzzification method and given by:

$$o_i^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) \quad (3.49)$$

where $\{p_i, q_i, r_i\}$ is the parameter set of the consequent part.

Layer 5: The single node in this layer computes the overall system output as the summation of all incoming signals:

$$o^5 = \sum_i \bar{w}_i f_i \quad (3.50)$$

ANFIS Learning

The basic learning rule of adaptive networks is based on the gradient descent method and the chain rule. Define the measure of error for the k^{th} input as

$$E_k = \frac{1}{2} \sum (d_k - o_k)^2 \quad (3.51)$$

where d_k is the desired system output and o_k is the actual system output.

In order to develop a learning procedure that implements gradient descent in E over the parameter space, first we have to calculate the **error rate** $\frac{\partial E_k}{\partial \beta_i}$ for training data and for each node output. We calculate the partial derivatives of the error function (3.51) with respect to the parameters of the fuzzy system, which need to be tuned. In other words, for every parameter β_i we have to use the chain rule in order to calculate the partial derivatives.

$$\frac{\partial E_k}{\partial \beta_i} = \frac{\partial E_k}{\partial o_k} \frac{\partial o_k}{\partial \beta_i} = -(d_k - o_k) \frac{\partial o_k}{\partial \beta_i} \quad (3.52)$$

Now, the error rate for consequence parameters can be calculated as follows:

$$\frac{\partial E}{\partial \beta_c} = \frac{\partial E}{\partial o_5} \frac{\partial o_5}{\partial o_4} \frac{\partial o_4}{\partial \beta_c} \quad (3.53)$$

where β_c is the consequence parameter and o_i is the output of the i^{th} layer.

While the error rate for premise parameters can be calculated as follows:

$$\frac{\partial E}{\partial \beta_p} = \frac{\partial E}{\partial o_5} \frac{\partial o_5}{\partial o_4} \frac{\partial o_4}{\partial o_3} \frac{\partial o_3}{\partial o_2} \frac{\partial o_2}{\partial o_1} \frac{\partial o_1}{\partial \beta_p} \quad (3.54)$$

where β_p is the premise parameter and o_i is the output of the i^{th} layer.

Start by computing the first partial derivative of the error

$$\frac{\partial E}{\partial o_5} = -(d - o) \quad (3.55)$$

The derivation of $\frac{\partial o_j}{\partial o_{j-1}}$, $j=5,4,3,2$ is calculated as follows:

$$\frac{\partial o_5}{\partial o_4} = \frac{\partial (\sum \bar{w}_j f_j)}{\partial (\bar{w}_i f_i)} = 1 \quad (3.56)$$

where \bar{w}_i is the normalized firing strength of the i^{th} rule

$$\frac{\partial o_4}{\partial o_3} = \frac{\partial (\bar{w}_i f_i)}{\partial (\bar{w}_i)} = f_i \quad (3.57)$$

where i is the number of the corresponding rule in the 3rd layer

$$\frac{\partial o_3}{\partial o_2} = \frac{\partial}{\partial w_i} \left(\frac{w_i}{\sum_{j=1}^n w_j} \right) = \frac{\sum_{j=1}^n w_j - w_i}{(\sum_{j=1}^n w_j)^2} \quad (3.58)$$

where i is the number of the corresponding rule in the 2nd layer, and n is the total number of rules in the system.

$$\frac{\partial o_2}{\partial o_1} = \frac{\partial}{\partial A_m} \left(\prod_{A_j \in \mathfrak{R}(A_m)} A_j \right) = \prod_{A_j \in \mathfrak{R}(A_m), A_j \neq A_m} A_j \quad (3.59)$$

where $A_j \in \mathfrak{R}(A_m)$ denotes the fuzzy sets, which make the premise part of the rule containing fuzzy set A_m

We can aggregate equations (3.55)-(3.59) to get a compact form of the partial derivatives.

For the consequence parameter β_c :

$$\frac{\partial E}{\partial \beta_c} = -(d - o) \frac{\partial o_4}{\partial \beta_c} \quad (3.60)$$

And for the premise parameter β_p of the membership function of linguistic label A_m :

$$\frac{\partial E}{\partial \beta_p} = -(d - o) f_i \frac{\sum_{j=1}^n w_j - w_i}{(\sum_{j=1}^n w_j)^2} \prod_{A_j \in \mathfrak{R}(A_m), A_j \neq A_m} A_j \frac{\partial o_1}{\partial \beta_p} \quad (3.61)$$

Now, calculate the partial derivatives of the output functions with respect to their parameters:

$$\frac{\partial o_4}{\partial p_i} = \frac{\partial}{\partial p_i} (\bar{w}_i f_i) = \frac{\partial}{\partial p_i} (\bar{w}_i (p_i x + q_i y + r_i)) = \bar{w}_i x \quad (3.62)$$

$$\frac{\partial o_4}{\partial q_i} = \frac{\partial}{\partial q_i} (\bar{w}_i f_i) = \frac{\partial}{\partial q_i} (\bar{w}_i (p_i x + q_i y + r_i)) = \bar{w}_i y \quad (3.63)$$

$$\frac{\partial o_4}{\partial r_i} = \frac{\partial}{\partial r_i} (\bar{w}_i f_i) = \frac{\partial}{\partial r_i} (\bar{w}_i (p_i x + q_i y + r_i)) = \bar{w}_i \quad (3.64)$$

where i is the number of the corresponding rule.

$$\frac{\partial o_1}{\partial a_{ij}} = \frac{\partial}{\partial a_{ij}} \left(\exp \left(- \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} \right) \right) = 2 \frac{\left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} b_{ij} \exp \left(- \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} \right)}{a_{ij}} \quad (3.65)$$

$$\frac{\partial o_1}{\partial b_{ij}} = \frac{\partial}{\partial b_{ij}} \left(\exp \left(- \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} \right) \right) = -2 \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} \ln \left(\frac{x-c_{ij}}{a_{ij}} \right) \exp \left(- \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} \right) \quad (3.66)$$

$$\frac{\partial o_1}{\partial c_{ij}} = \frac{\partial}{\partial c_{ij}} \left(\exp \left(- \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} \right) \right) = 2 \frac{\left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} b_{ij} \exp \left(- \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}} \right)}{x-c_{ij}} \quad (3.67)$$

where i is the number of the corresponding rule and j is the number of the corresponding linguistic variable in the rule.

To simplify the premise parameters updating formulas, assume that $G_{ij} = \left(\frac{x-c_{ij}}{a_{ij}} \right)^{2b_{ij}}$ then

$$\frac{\partial o_1}{\partial a_{ij}} = 2 \frac{b_{ij}}{a_{ij}} G_{ij} \exp(-G_{ij}) \quad (3.68)$$

$$\frac{\partial o_1}{\partial b_{ij}} = -2 \ln \left(\frac{x-c_{ij}}{a_{ij}} \right) G_{ij} \exp(-G_{ij}) \quad (3.69)$$

$$\frac{\partial o_1}{\partial c_{ij}} = 2 \frac{b_{ij}}{x-c_{ij}} G_{ij} \exp(-G_{ij}) \quad (3.70)$$

Different learning schemes have been presented by Jang to update the parameters of ANFIS where the Gradient Decent method (GD) is used in most of them. GD could be used to update the parameters in both the premise and consequent parts. Another version is similar to the first one but the Least squares estimation (LS) is applied once to get the initial values as a pre-processing phase. A hybrid version has been introduced that employs the GD to update the nonlinear parameters in the premise part while using least square estimation to renew the linear weights in the consequent part. Lastly, using sequential least square estimations such as: Recursive Least Squares (RLS) and Extended Kalman filtering (EKF) to update ANFIS parameters [59].

The ANFIS network performance is considered to be decent though it suffers from some significant drawbacks such as the *curse of dimensionality*. In ANFIS, the required number of rules relies directly on the length of the input vector and is calculated as $\text{Rules} = (\text{MF})^{\text{input}}$ where MF denotes the number of membership functions representing the input fuzzy partitions. In cases where number of inputs is high, the parameters, which need to be evaluated, will consequently increase. Hence, the least square estimation will have to deal with very large matrices and time consuming calculations [21, 22].

3.4 Cluster Analysis

Large amount of information is encountered daily for storage or further investigation and management. A dynamic attitude in managing these data is to classify or group them into a set of categories or clusters. Clustering is the process of grouping vectors into classes based on their similar features employing a self-organising mode. A cluster is a group of objects that are more alike to one another than to members of other groups. The aim of cluster analysis is the classification of objects according to similarities and organizing data into groups. The similarity between elements should be comprehended as a mathematical term, measured in precise sense. In metric spaces, the term “similarity” in many cases is considered as a *distance norm* from a data vector to a prototype object called the centre [60].

Clustering proved to be an effective approach to detect the underlying structure in complex nonlinear data. Based on the clustering output, the space of input would be partitioned into subsets and a simple model estimates each subset. The data used could be quantitative (numerical) and/or qualitative (categorical). Clustering techniques are used in various applications such as pattern recognition, model reduction and optimization.

Clustering methods are unsupervised networks where the class identifiers are unavailable.

In general, major clustering methods can be classified into five categories.

- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Methods

In this research, the partitioning methods are discussed. Since clusters are basically subsets of the original data set, clustering methods could be classified according to whether the obtained subsets are fuzzy or crisp. Based on that, the partitioning methods are divided into: Hard and Soft clustering. Hard clustering methods are based on classical set theory, thus forming an exclusive crisp partition of the given data where any object either does or does not belong to a cluster. On the other hand, Soft or Fuzzy clustering methods allow objects to belong to several clusters simultaneously, with different degrees of membership. This creates the concept of fuzzy boundaries, which differs from the traditional concept of well-defined boundaries [61]. In actual conditions, fuzzy clustering is more accepted than hard clustering, as objects on the boundaries are not forced to fully belong to one of the classes, but rather are assigned gradual membership degrees between 0 and 1. Moreover, in fuzzy clustering, instead of determining whether or not an event occurs, as is the case with probability, fuzziness measures the degree to which an event occurs.

3.4.1 Fuzzy C-Means Clustering

Fuzzy C-Means (FCM) is one of the most commonly used clustering techniques in different problems. FCM employs two straightforward statistical features, namely the mean and standard deviation. This method was initiated by Dunn in 1973 [62] and then improved and popularised by Bezdek in 1981 [63].

FCM produce their results in the form of a fuzzy partition matrix and prototype locations representing the cluster centres. A membership value is assigned to each data point corresponding to the distance between the cluster center and the data point. If the data point is closer to the center of the cluster, its membership value increases. A comparison between the final results of hard and fuzzy clustering is shown in Fig. 3.11.

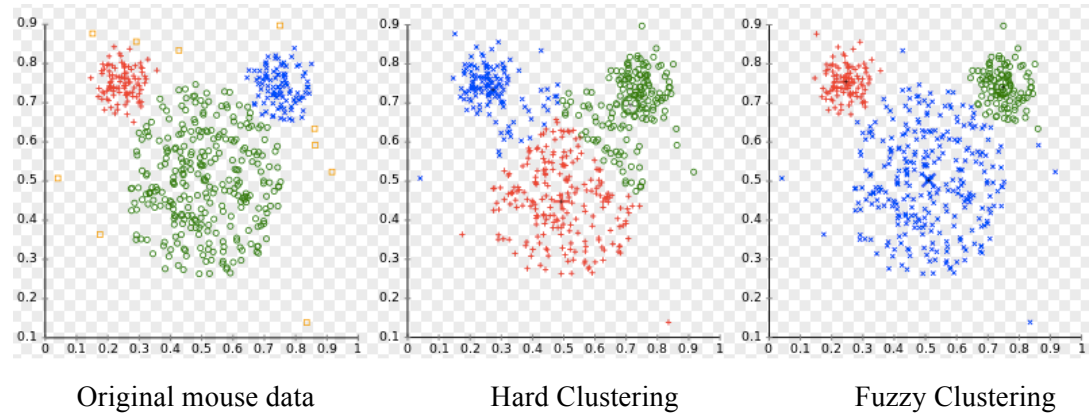


Fig. 3.11 Comparison between hard and fuzzy clustering

This iterative algorithm uses a predetermined number of clusters where no optimization is done to change the integer [64]. After each iteration, the membership matrix μ_{ij} and accordingly the cluster centers v_j are updated using the two following formulas:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{D_{ij}}{D_{ik}} \right)^{\frac{2}{m-1}}}, \forall i = 1, 2, \dots, n \text{ and } \forall j = 1, 2, \dots, c \text{ and } m \in [1, \infty) \quad (3.71)$$

$$v_j = \frac{\sum_{i=1}^n (\mu_{ij})^m x_i}{\sum_{i=1}^n (\mu_{ij})^m}, \forall j = 1, 2, \dots, c \quad (3.72)$$

Where n is the number of input, c is the number of clusters, D_{ij} represents the distance between i input and j cluster center, and m is the fuzziness index.

Algorithmic steps of the FCM

Let $X = \{x_1, x_2, \dots, x_n\}$ be the a set of data points and $V = \{v_1, v_2, \dots, v_c\}$ be the initial set of cluster centers where c is the selected number of clusters. Also assume the fuzziness index m to be equal to 2. The main objective of the fuzzy c-means algorithm is to minimise the function:

$$J(U, V) = \sum_{i=1}^n \sum_{j=1}^c (\mu_{ij})^2 \|x_i - v_j\|^2 \quad (3.73)$$

where $\|x_i - v_j\|$ is the Euclidean distance.

Step 1: Randomly select c cluster centers

Step 2: Calculate the fuzzy membership matrix μ_{ij} using the formula

$$\mu_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^2} \quad (3.74)$$

Step 3: Compute the new fuzzy centers using the equation

$$v_j = \frac{\sum_{i=1}^n (\mu_{ij})^2 x_i}{\sum_{i=1}^n (\mu_{ij})^2}, \forall j = 1, 2, \dots, c \quad (3.75)$$

Step 4: Repeat steps 2 and 3 until the minimum of the objective function is reached

$$J(U, V) = \sum_{i=1}^n \sum_{j=1}^c (\mu_{ij})^2 \|x_i - v_j\|^2 \quad (3.76)$$

The FCM algorithm is widely used and well recognised in different disciplines for its advantages. In terms of coding implementation, it is relatively straightforward and employs an objective function that is intuitive and easy-to-grasp [65]. The algorithm has proven to be convergent with robust performance and delivers consistent membership values.

The method experiences some disadvantages. Firstly, the number of clusters has to be determined by the user. Also, the initialization of the centres has a direct impact on the final findings. So when the iterative algorithm is employed to minimise the objective function, it could get trapped in local minima instead of finding the global depending on the initial points used. Finally, the algorithm is sensitive to noise and outliers.

Chapter Four

COMPUTATIONAL INTELLIGENCE TECHNIQUES TO ASSESS SENSORY QUALITY OF MEAT USING FTIR SPECTROSCOPY

4.1 Introduction

In the past few decades, meat industry has enormously thrived, as demands for better food quality continues to grow on both international and domestic markets [66]. Interests in meat quality are driven by the need to supply the consumer with a consistent high quality product at an affordable price [67]. To realise such need, it is a crucial element within the meat industry to accurately assess meat quality attributes by improving modern techniques for quality evaluation [68].

Beef is one of the commercially viable and widely consumed muscle foods throughout the world. Although it is a good food source for proteins and other essential nutrients, it is also an ideal substrate for the growth of both spoilage and pathogenic microorganisms. Spoilage occurs when the formation of off-flavours, off-odours, discoloration, or any other changes in physical appearance or chemical characteristics make the food unacceptable to the consumer. Changes in muscle food characteristics are due to native or microbial enzymatic activity or to other chemical reactions. The current practice to assure the safety of meat still relies on regulatory inspection and sampling regimes. This approach, however, seems inadequate because it cannot sufficiently guarantee consumer protection, since 100% inspection and sampling is technically, financially and logistically impossible. Additionally, although more than 50 chemical, physical and microbiological methods have been proposed for the detection and measurement of bacterial safety or spoilage in meat, most of them are time-consuming and provide retrospective information [69].

Meat industry however needs rapid analytical methods for quantification of these indicators in order to determine suitable processing procedures for their raw material and to predict the remaining shelf life of their products [70]. The development of non-destructive sensing technologies to detect spoilage bacteria as well as pathogenic bacteria with a high degree of dependency in food products is very desirable. Various rapid, non-invasive methods based on analytical instrumental techniques, such as Fourier transform infrared spectroscopy (FTIR) [71], Raman spectroscopy [72], and Electronic nose technology [73] have been researched for

their potential in assessing meat quality. In recent years, spectral imaging (i.e., hyperspectral and multispectral) has been also considered as an alternative tool for safety and quality inspection of various agricultural products [74]. This technique integrates the conventional imaging and spectroscopy technique to attain simultaneously both spatial and spectral information from the target product. The “mechanism” of these approaches is based on the assumption that the metabolic activity of micro-organisms on meat results in biochemical changes, with the simultaneous formation of metabolic by-products, which may contribute to the spoilage phenomenon. The quantification of these metabolic activities corresponds to a unique “signature”, providing thus information about the type and rate of spoilage [75].

The huge amount of information provided by analytical sensors/devices requires an advanced data analysis approach. This has been achieved through the integration of modern analytical platforms with computational and chemometric techniques [76]. Multivariate statistical analyses (*e.g.*, partial least square (PLS) regression, discriminant function analysis (DFA), cluster analysis) have resulted in the development of decision support systems for timely determination of safety/quality of meat products [77]. Considering that microbial meat spoilage is a complex process, which involves growth of microorganisms during storage, their spectra contain highly non-linear characteristics. Hence, linear-based techniques might not provide a complete solution to such complex identification/classification problem [78]. Neural networks (NNs) have gained much interest in predictive engineering and quantitative modelling due to their flexibility and high accuracy as compared to other modelling techniques (*e.g.*, statistical models). In comparison to other NN-based application areas, the field of food science is still in an early development stage. Recently, advanced NN algorithms have shown promising results in applications such as growth parameter estimation of microorganisms [79]. NNs usually require a large number of neurons for solving the majority of approximation problems and are prone to dimensionality problems, as each single neuron-node cannot define a multi-dimensional hyper-sphere of the input domain. Although fuzzy logic systems, provide such input space mapping, they do not have learning ability, thus it is difficult to analyse complex systems without prior and accurate knowledge on the system being analysed [80].

To overcome the limitations of NNs and fuzzy systems, neuro-fuzzy approaches have attracted growing interest of researchers in various scientific and engineering areas. Two specific case studies have been considered in this research and are presented with all necessary details in the current and next chapters. In the first case, an Adaptive Fuzzy Logic System (AFLS) model has been developed for the detection of beef spoilage using FTIR spectral data, while in the second case, an Adaptive Fuzzy Inference Neural Network (AFINN) has been used, for the first time according to literature, to associate meat spoilage with spectral data, acquired by multispectral imaging techniques.

To address the rapid and non-destructive detection of meat spoilage microorganisms during aerobic storage at chill and abuse temperatures, Fourier transform infrared (FTIR) spectroscopy with the aid of a neurofuzzy identification model has been considered in this chapter. Datasets related to FTIR and imaging spectral information as well as the correlated microbiological analysis (i.e. total viable counts - TVC) from meat samples, were provided by Agricultural University of Athens, Greece. Fresh beef fillets were packaged under aerobic conditions and left to spoil at (0, 5, 10, 15, 20 °C), for up to 350 h. FTIR spectra were collected directly from the surface of meat samples, whereas TVCs of bacteria were obtained via standard plating methods. Sensory evaluation was performed during storage and samples were attributed into three quality classes, namely fresh, semi-fresh, and spoiled [81]. The proposed AFLS model has been utilised to simultaneously classify beef samples to one of three quality classes, based on their biochemical profile provided by the FTIR spectrometer and predict TVCs on meat surface. Results from AFLS scheme are compared against models based on ANFIS, multilayer neural networks (MLP), and PLS regression models. Such comparison is considered as an essential practice, as we have to emphasise the need of induction to the area of food microbiology, advanced learning-based modelling schemes, which may have a significant potential for the rapid and accurate assessment of meat spoilage.

4.2 FTIR Spectroscopy in Food Quality Analysis

FTIR spectroscopy has been used extensively by chemists to identify compounds in a wide variety of applied fields. This motivation is justified from the fact that FTIR is a rapid, inexpensive and sensitive technology that rapidly allows real-time measurements at all stages of production without requiring special skills from users. Moreover, it has been recognized as a powerful tool when coupled to chemometric techniques, and is widely utilised for rapid quality control of numerous foodstuffs since it provides information from complex spectra about the composition of food components. In fact, for any individual sample, FTIR spectroscopy provides information on fundamental vibration and stretching of molecules exhibited under infrared light in the spectral region between 4000 and 700 cm^{-1} . It provides a characteristic spectrum that is the result of absorption by various chemical constituents, providing thus a “fingerprint” of each sample [82].

Over the last few years, FTIR has been considered as a very important tool in food analysis including authenticity and adulteration. Nutrient determination is time consuming and not appropriate for routine application in the food industries. FTIR was able to determine omega-6 and omega-3 fatty acids in pork adipose tissue [83]. It has been used to investigate the

influence of heating rates and different heating temperatures on protein denaturation in beef [84], as well as to study the influence of ageing and salting on uncooked and cooked pork [85]. In addition, it has been considered as a fast and non-destructive technique for the detection and quantification of pork in beef meatball formulation for Halal verification purposes. The spectral bands associated with pork fat (PF), beef fat (BF), and their mixtures in meatball formulation, were scanned, interpreted, and identified by relating them to those spectroscopically representative to pure PF and BF [86]. The quality of oil, an essential ingredient of the food processing industry has also been investigated through the use of FTIR for the accurate quantification of the moisture in edible oils [87]. Although, the vast majority of FTIR-based applications for classification and quantification purposes utilise liquid and solid phase samples, very few applications based on gaseous phase samples have been reported. A FTIR spectroscopy-based olfactory sensing system has been investigated in sensing the volatiles collected from the headspace of a meat package for the detection of pathogen contamination (*Salmonella typhimurium*) in the packed meat [88].

The application of chemometric techniques to associate FTIR spectral data with meat spoilage is not new and it has been tackled in the past [89]. In that approach, emphasis was given only to the detection of bacterial spoilage, in terms of microbiological analysis, whereas no attempt was made to associate spectral data with quality classes defined by sensory assessment of the samples. FTIR spectral data collected directly from the surface of meat had been verified that they could be used as biochemical interpretable “signatures”, in an attempt to obtain information on early stage detection of microbial spoilage of chicken breast and rump steaks [90]. An MLP-based NN has been also explored in an attempt to correlate spectral data from FTIR spectroscopy analysis with beef spoilage and its associated total viable bacteria counts-TVC [81]. A rapid and non-destructive determination of pork storage time associated with its freshness has been considered with the aid of FTIR and MLP networks [91], while the spoilage of fresh minced pork was examined through the use of FTIR and PLS models [92]. Recently, the potential of mid infrared spectroscopy to determine microorganisms’ spoilage on the surface of chicken breast fillets has been explored. PLS regression was carried out to predict TVCs, *Pseudomonas*, *Enterobacteriaceae* and *Brochothrix thermosphacta* microbial counts from related spectral data [83].

4.3 FTIR Sampling and Analysis

The FTIR experimental case was performed at the Laboratory of Microbiology and Biotechnology of Foods, at the Agricultural University of Athens, Greece. A detailed description of the experimental methodology, as well as the related microbiological analysis

of the meat samples, is described in [81]. Briefly, the samples were prepared by cutting fresh pieces of beef into small portions (40 mm wide \times 50 mm long \times 10 mm thick) and then portions placed onto Petri dishes and stored at (0, 5, 10, 15 and 20 °C) in high-precision incubation chambers for a total period of 350 h, taking into consideration the storage temperature, until spoilage was apparent [81]. For the purposes of FTIR spectral measurements, a thin slice of the aerobic upper surface of the beef fillet was isolated and used for additional analysis. In total, 74 FTIR spectra were produced through the use of a ZnSe 45° ATR (Attenuated Total Reflectance) crystal on a Nicolet 6700 FTIR Spectrometer, as shown in Fig. 4.1. The samples were placed on the ZnSe ATR crystal so that the aerobic upper surface of the meat was in intimate contact with the crystal. The sample then was pressed with a gripper so as to have better possible contact with the crystal. The spectrometer was programmed with Omnic Software to collect spectra over the wave number range 4000 to 400 cm^{-1} , whilst the scans per measurement were 100 with a resolution of 4 cm^{-1} , resulting in a total integration time of 2 min. The ZnSe ATR crystal was capable of 12 external reflections, with the evanescent field affecting a depth of 1.01 μm . Each sample was analysed in duplicate and results are displayed as mean value of both measurements. Reference spectra were obtained by collecting a spectrum from the cleaned blank crystal prior to the presentation of each sample replicate. At the end of each sampling, the crystal surface was cleaned with detergent, washed with distilled water, dried with lint-free tissue, cleaned with ethanol and finally dried with lint-free tissue at the end of each sampling interval.

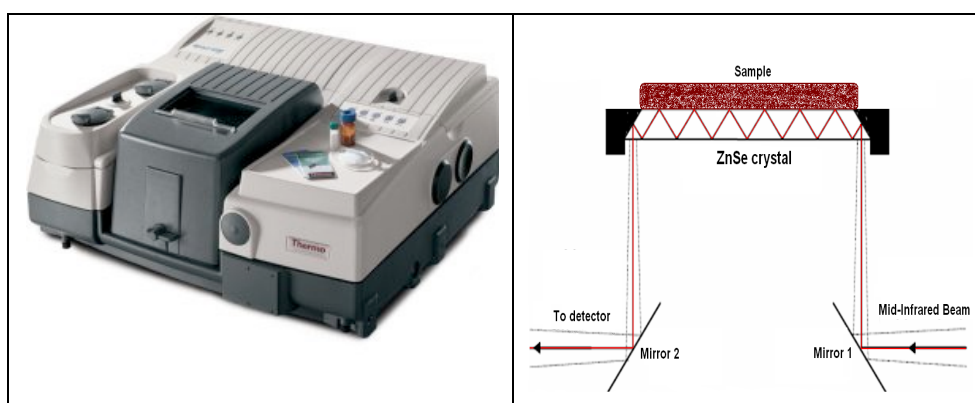


Fig. 4.1 FTIR spectroscopy device / principles

Spectra collected over the specific wave-number range 1800–1000 cm^{-1} , which is considered as the area where spoilage signals are expected to be detected [81].

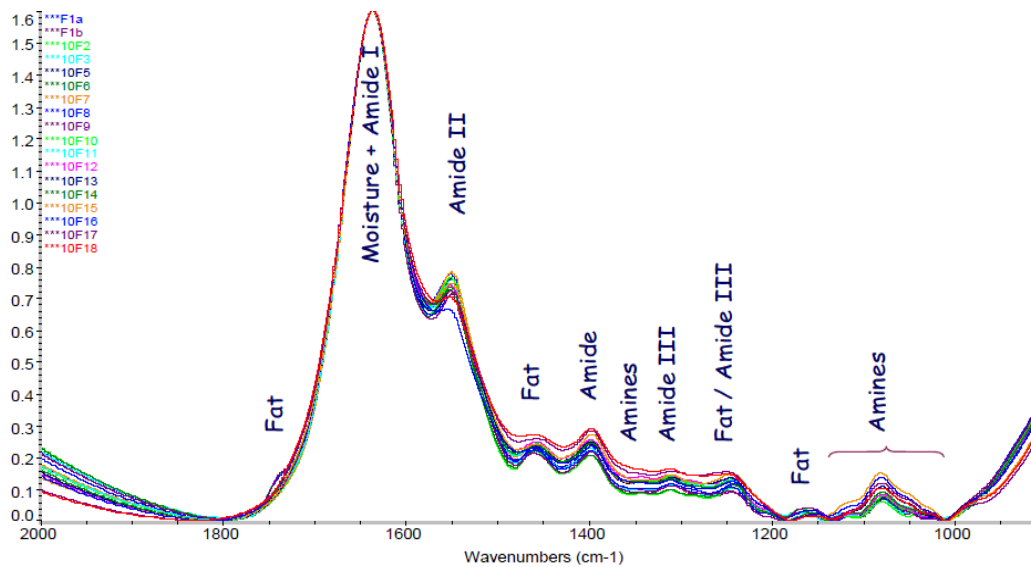


Fig. 4.2 FTIR spectra collected from beef samples stored at 10°C

Typical FTIR spectral data collected from fresh and spoiled beef fillet samples stored at 10°C for 6 days are shown in Fig. 4.2. Different colours represent samples collected at various sampling times. For example, sample 10F2 has been collected after 8h storage, while 10F18 is associated with a sample with storage of 148h. Two curves associated to F1a and F1b are considered as measurements that correspond to the initialisation of the FTIR procedure, at $t = 0$ sec [81]. Comparison of those FTIR spectra can reveal knowledge on particular biochemical changes taking place during beef spoilage, at various temperatures. For example, the highest peak 1640cm^{-1} is due to the presence of moisture with an essential contribution from amide I in the beef sample. The second peak at 1550cm^{-1} is due to the absorbance of amide II, while peaks at 1460, 1240, and 1175cm^{-1} can be credited to fat. These FTIR spectra were initially subjected to smoothing according to the Savitzky-Golay algorithm [81].

In parallel, microbiological analysis was performed, and resulting growth data from plate counts were \log_{10} transformed and fitted to the primary model of Baranyi & Roberts [93] in order to verify the kinetic parameters of microbial growth (maximum specific growth rate and lag phase duration). The growth curves of total viable counts (TVC) for beef fillet storage at different temperatures under aerobic conditions are illustrated in Fig. 4.3.

Analysis specified that the total microflora (TVC) ranged from $2.9\text{--}3.3\log_{10}\text{ cfu cm}^{-2}$ at the beginning of storage (fresh samples), to $8.7\text{--}9.4\log_{10}\text{ cfu cm}^{-2}$ for samples characterised as spoiled [81]. This finding is consistent with an indication that the population threshold that depicts the shift of a sample from fresh to semi-fresh and then from semi-fresh to spoiled is temperature dependant.

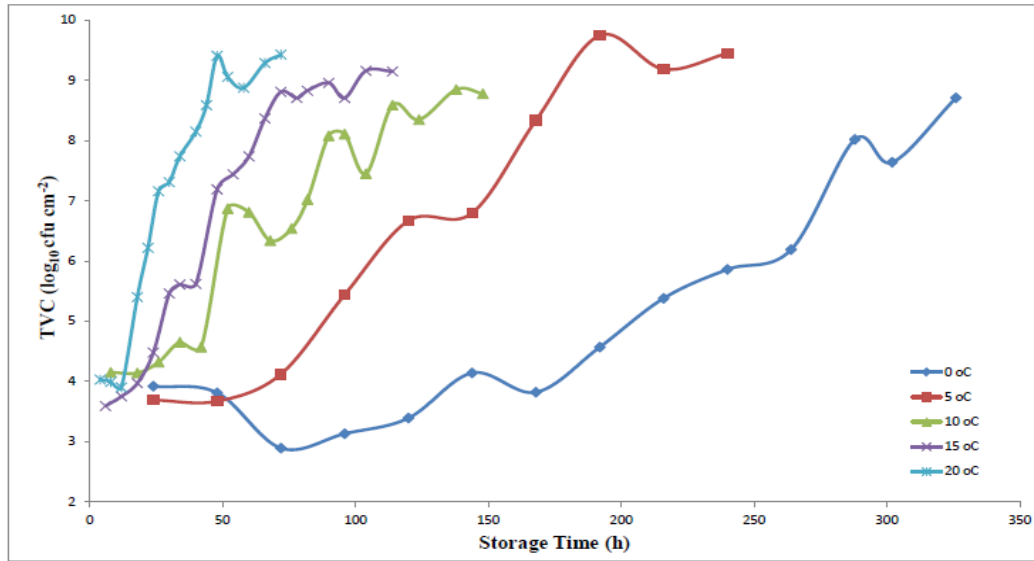


Fig. 4.3 Population dynamics of TVC at various temperatures for beef samples

Additionally, sensory evaluation of meat samples was performed during storage, based on the perception of colour and smell before and after cooking. Each sensory attribute was assigned to a three-point scale corresponding to: 1=fresh; 2=semi-fresh; and 3= spoiled. In total, 74 meat samples were evaluated by a sensory panel and classified into the selected three groups as fresh (n = 24), semi-fresh (n = 16), and spoiled (n = 34) [4.16].

| PCs | PCA | | |
|-----|------------|---------|--------------|
| | Eigenvalue | Prop. % | Cum. prop. % |
| 1 | 190.080 | 70.925 | 70.925 |
| 2 | 48.083 | 17.941 | 88.867 |
| 3 | 12.754 | 4.759 | 93.626 |
| 4 | 7.215 | 2.692 | 96.318 |
| 5 | 5.194 | 1.938 | 98.256 |
| 6 | 1.807 | 0.674 | 98.930 |
| 7 | 1.070 | 0.399 | 99.329 |

Table 4.1 Eigenvalues and proportion of the first seven PCs for PCA scheme

Due to the nature of FTIR spectral data, the use of principal component analysis (PCA) in reducing the level of input dimensionality with the minimum information lost is required. A principal component analysis has been applied on those FTIR spectral data used for training purposes. This choice was initiated mainly by two reasons. The original FTIR spectral data is considered as a high-dimensional problem, thus it could be prohibited for utilising a learning-based model (*i.e.* excessive number of input variables). Secondly, the strong correlation among the FTIR variables (*i.e.* wave-numbers), would deteriorate seriously the modelling procedure. PCA scheme was implemented in MATLAB, with the aid of PLS_Toolbox (ver. 7.5, Eigenvector.com). For this particular experimental case study, although the total variance

(100%) of the dataset was explained by 34 principal components (PCs), only the first five PCs were associated with the 98.25% of the total variance, as shown in Table 4.1. The variability (%) of the first three PCs is dominant to the overall contribution and this is also illustrated from a visualisation of the first three orthonormal principal component coefficients for each variable, and the principal component scores for each observation in a single plot, as shown at Fig. 4.4.

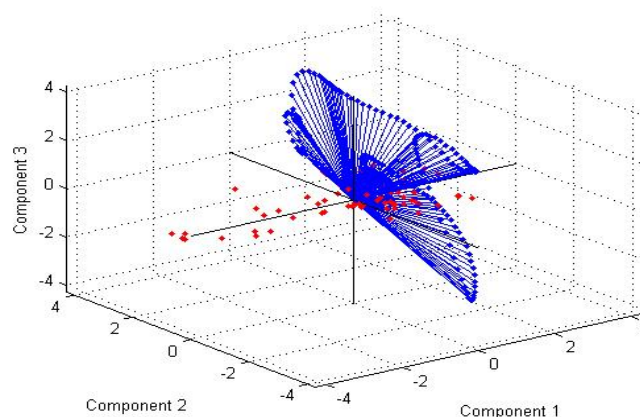


Fig. 4.4 3-D plot for the first three principal components

Thus, the first five principal components from the PCA scheme were extracted and utilised as inputs to the various simulation models developed for this specific case study.

4.4 Adaptive Fuzzy Logic System (AFLS)

With the continuously growing demand for models for complex systems inherently associated with nonlinearity, high-order dynamics and imprecise measurements, there is need for a relevant modelling environment. During the last decade, neurofuzzy network (NF) approaches have gained considerable interest for solving real world problems, including modelling of highly complex systems and pattern recognition. Extensive experimentation has demonstrated that the class of feed-forward NF systems exhibits significant advantages compared to the NN models [94]. NNs are well-known to act as global models, where training is performed on the entire pattern range. In contrast, owing to the partition of the input space, NF models perform a fuzzy blending of local models in space. As a result, faster convergence is achieved during learning for a specific task. Additionally, by utilising learning-based training algorithms for the tuning of fuzzy logic parameters, the efficiency of function approximation can be largely improved. Examples of NF systems as modelling structures include schemes such as ANFIS, Fuzzy RBF, and Fuzzy Adaptive learning Control Network (FALCON) [95].

4.4.1 AFLS Architecture

A Fuzzy Logic System (FLS) is a system that utilises fuzzy set theory and its operations to solve a given problem. A specific type of FLS with fuzzifier and defuzzifier components is used throughout this study. Various methods have been used in defuzzification section, such as “centre average”, “centroid of area”, *etc.* The “centroid of area” approach is an optimal answer to defuzzification because it uses all available information to compute the output. One major problem, however, with the centroid defuzzifier is its intensive computation. The “centre average” (CA) defuzzifier, on the other hand, is more efficient in terms of implementation. Its main disadvantage is that it suffers from not utilising the entire shape of the consequent membership function. The output of a CA defuzzifier is still the same, regardless of whether the shape is narrow or wide [96]. In general, an adaptive fuzzy logic system is a FLS where its rules are derived and extracted from given training data. Conventional fuzzy rule-table approaches usually utilise the “look-up table” concept. In those models, an input space is divided into $K_1 \times K_2 \times \dots \times K_n$ fuzzy subspaces, where K_i , $i = 1, 2, \dots, n$ is the number of fuzzy subsets for the i^{th} input variable [97]. As one fuzzy rule is normally assigned for each one of these subspaces, their main drawback is that the number of fuzzy rules increases exponentially with respect to the number of inputs n .

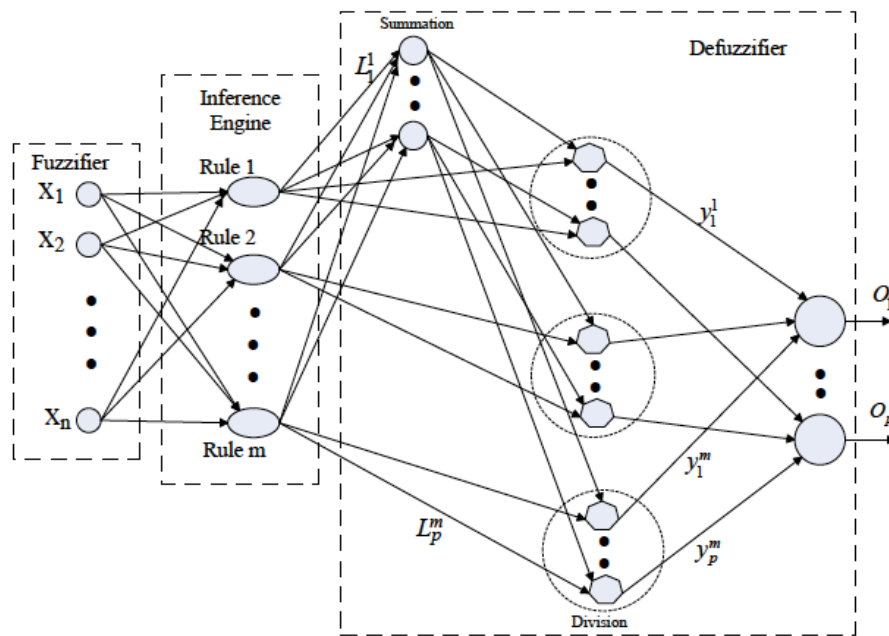


Fig. 4.5 AFLS Architecture

The Adaptive Neurofuzzy Inference System (ANFIS) is a classic example of such approach, where the number of fuzzy rules is related to the number of input variables as well as the number of membership functions for each input. In the case of AFLS, the number of

memberships for each input variable is directly associated to the number of rules, hence, the “curse of dimensionality” problem is significantly reduced. Since we have general ideas about the structure and effect of each rule, it is straightforward to effectively initialise each rule. This is a tremendous advantage of AFLS over its NN counterpart. The “centroid” defuzzifier however cannot be used also due to the presence of gradient-descent learning algorithm. The proposed AFLS scheme consists of an alternative defuzzification approach, the area of balance (AOB), and its structure is shown in Fig. 4.5 [98]. In this architecture, the fuzzy basis layer consists of fuzzy basis nodes for each rule. A fuzzy basis node has the following form:

$$\varphi_m(\bar{x}) = \frac{\mu_m(\bar{x})}{\sum_{l=1}^L \mu_l(\bar{x})} \quad (4.1)$$

where $\varphi_m(\bar{x})$ is a fuzzy basis node for rule m and $\mu_m(\bar{x})$ is a membership value of rule m .

Since a product-inference is utilised, the fuzzy basis node $\mu_m(\bar{x})$ is in the following form:

$$\mu_m(\bar{x}) = \prod_{i=1}^n \mu_{F_i^m}(x_i) \quad (4.2)$$

where $\mu_{F_i^m}(x_i)$ is the membership value of the i^{th} input of rule m . In the proposed scheme, a “Gaussian-shape” membership function has been employed, thus $\mu_{F_i^m}(x_i)$ has the following form:

$$\mu_{F_i^m}(x_i) = \exp \left[-\frac{(x_i - c_i^m)^2}{2(b_i^m)^2} \right] \quad (4.3)$$

where c_i^m and b_i^m are the centre and spread parameters of the membership function i^{th} input of the m^{th} rule.

The “centroid of area” defuzzification method returns the centroid of the area formed by the consequent membership function, the membership value of its rules and the max-min or max-product inference. In the case of a discrete universe, the centroid calculation yields

$$y = \frac{\sum_{q=1}^Q \mu_y(y_b^q) y_b^q}{\sum_{q=1}^Q \mu_y(y_b^q)} \quad (4.4)$$

where Q is a number of quantisation levels of the output. The higher Q is the finer y will be. The computational cost increases analogous to Q . However, since the method provides good performance, its main characteristics, such as centre of gravity and use of the shape of membership function, have been adopted in the design of the proposed defuzzification approach.

AFLS's overall output utilises Kosko's method with product inference [99]. The proposed defuzzification method can be explained by the following mass-less beam example. Let us consider the density (D), which is defined as mass (M) per unit volume (V).

$$D = \frac{M}{V} \quad (4.5)$$

Under the assumption that we use the same material and all shapes have the same thickness, T, then

$$M = ATD \quad (4.6)$$

where A is an area and T is a thickness. Let us assume for simplicity, that the shape of the membership function used in the consequent part has a symmetric triangular form.

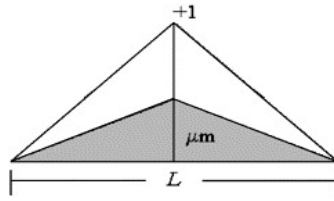


Fig. 4.6 Triangular shape membership function

The centre of gravity will pass through the halfway point of the base of that shape. For example, if we use a triangular shape and product-inference as a t-norm, then the shape of the consequent part of rule m will be shown as in Fig. 4.6. If we could consider the consequent part of each rule placed on the massless beam having the pivot point at origin, then such visualisation is shown in Fig. 4.7.

Then,

$$F = M_1g + M_2g + M_3g = (M_1 + M_2 + M_3)g \quad (4.7)$$

For balance

$$Fy = M_1gy_b^1 + M_2gy_b^2 + M_3gy_b^3 = (M_1y_b^1 + M_2y_b^2 + M_3y_b^3)g \quad (4.8)$$

$$y = \frac{(M_1y_b^1 + M_2y_b^2 + M_3y_b^3)g}{F} = \frac{(M_1y_b^1 + M_2y_b^2 + M_3y_b^3)g}{(M_1 + M_2 + M_3)g} \quad (4.9)$$

Assume that D and T in Eq. 4.6 is the same, thus,

$$y = \frac{(A_1y_b^1 + A_2y_b^2 + A_3y_b^3)TD}{(A_1 + A_2 + A_3)TD} = \frac{(A_1y_b^1 + A_2y_b^2 + A_3y_b^3)}{(A_1 + A_2 + A_3)} \quad (4.10)$$

The calculation of area A will be depended upon the type of membership function used. Under the assumption of symmetric shape, this method will have comparable capability with the centroid calculation method to approximate the output from the fuzzy set in the consequent part. By utilising the triangle shape as a membership function and the usage of max-product inference (*i.e.* Larsen logic), the shaded area A will be derived as:

$$A_m = \frac{1}{2} \mu_m L_m \quad (4.11)$$

Deriving from Eq. 10, the output, y will be

$$y = \frac{(\mu_1 L_1 y_b^1 + \mu_2 L_2 y_b^2 + \mu_3 L_3 y_b^3)}{(\mu_1 L_1 + \mu_2 L_2 + \mu_3 L_3)} \quad (4.12)$$

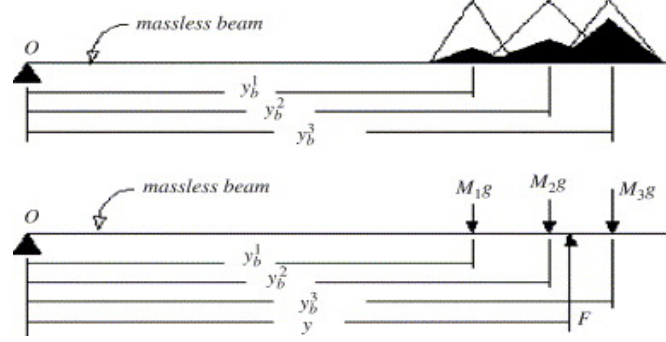


Fig. 4.7 Consequent fuzzy set placed on mass-less beam

In general form, the calculation of the output, y will be

$$y_p = \frac{\sum_{m=1}^M \mu_m L_p^m y_p^m}{\sum_{m=1}^M \mu_m L_p^m} \quad (4.13)$$

where

- y_p : the p^{th} output of the network
- μ_m : the membership value of the m^{th} rule
- L_p^m : the spread parameter of the membership function in the consequent part of the p^{th} output of the m^{th} rule
- y_p^m : the centre of the membership function in the consequent part of the p^{th} output of the m^{th} rule.

4.4.2 AFLS learning phase procedure

The learning algorithm of AFLS involves the use of the backpropagation/gradient descent (BP/GD) method to identify the various parameters. During, the backward “training” pass, the error signals are calculating recursively from the output layer backward to the hidden (*i.e.* rules) layer, and parameters at defuzzification and fuzzification parts are fine-tuned.

The objective function for training is defined as:

$$J = \sum_{k=1}^K J_k \quad (4.14)$$

where K is the number of training patterns and J_k is the sum of squared error for the k^{th} pattern. Then, J_k is defined as:

$$J_k = \frac{1}{2} \sum_{p=1}^P (y_p(\bar{x}_k) - d_p(\bar{x}_k))^2 \quad (4.15)$$

where P is the number of outputs and d_p is the desired response of the p^{th} output. Variable $y_p(\bar{x}_k)$ is defined as in Eq. 4.13. The update equation of y_p^m is as in the form:

$$y_p^m(n+1) = y_p^m(n) + m_y [y_p^m(n) - y_p^m(n-1)] - \eta_y \frac{\partial J}{\partial y_p^m} \Big|_n \quad (4.16)$$

where

$$\frac{\partial J}{\partial y_p^m} = \frac{\partial J}{\partial J_m} \frac{\partial J_m}{\partial e_p} \frac{\partial e_p}{\partial y_p} \frac{\partial y_p}{\partial y_p^m} \quad (4.17)$$

$$= \sum_{k=1}^K \{ (y_p^k - d_p^k) \frac{\mu_m^k L_p^m}{\sum_{m=1}^M \mu_m^k L_p^m} \} \quad (4.18)$$

where y_p^k is the p^{th} output of the network corresponding to the k^{th} pattern in the training data, d_p^k is the p^{th} desired output of the k^{th} pattern and μ_m^k is the membership value of the m^{th} rule corresponding to the k^{th} pattern in the training data. The update equation of L_p^m is in the following form:

$$L_p^m(n+1) = L_p^m(n) + m_L [L_p^m(n) - L_p^m(n-1)] - \eta_L \frac{\partial J}{\partial L_p^m} \Big|_n \quad (4.19)$$

where

$$\frac{\partial J}{\partial L_p^m} = \frac{\partial J}{\partial J_m} \frac{\partial J_m}{\partial e_p} \frac{\partial e_p}{\partial y_p} \frac{\partial y_p}{\partial L_p^m} \quad (4.20)$$

$$\frac{\partial J}{\partial L_p^m} = \sum_{k=1}^K \left[(y_p^k - d_p^k) \frac{\mu_m^k}{\left[\sum_{m=1}^M \mu_m^k L_p^m \right]} \{ y_p^m - y_p^k \} \right] \quad (4.21)$$

where y_p^m is interpreted as a centre of the membership function of the p^{th} output of the m^{th} rule in the consequent part of IF-THEN rule. The update equation of the centre parameter c_i^m is in the form:

$$c_i^m(n+1) = c_i^m(n) + m_c [c_i^m(n) - c_i^m(n-1)] - \eta_c \frac{\partial J}{\partial c_i^m} \Big|_n \quad (4.22)$$

where

$$\frac{\partial J}{\partial c_i^m} = \frac{\partial J}{\partial J_m} \frac{\partial J_m}{\partial e_p} \frac{\partial e_p}{\partial y_p} \frac{\partial y_p}{\partial \mu_m} \frac{\partial \mu_m}{\partial \mu_{F_i^m}} \frac{\partial \mu_{F_i^m}}{\partial c_i^m} + \dots \quad (4.23)$$

As, a Gaussian-based membership function has been adopted, then

$$\frac{\partial \mu_{F_i^m}}{\partial c_i^m} = \mu_{F_i^m} \frac{(x_i^k - c_i^m)}{b_i^{m^2}} \quad (4.24)$$

and

$$\frac{\partial \mu_m}{\partial \mu_{F_i^m}} = \prod_{\substack{j=1 \\ j \neq i}}^n \mu_{F_i^j} \quad (4.25)$$

Thus,

$$\frac{\partial J}{\partial c_i^m} = \sum_{k=1}^K \left[\left\{ \sum_{p=1}^P (y_p^k - d_p^k) \frac{L_p^m [y_p^m - y_p^k]}{\sum_{m=1}^M \mu_m^k L_p^m} \right\} \mu_m^k \frac{(x_i^k - c_i^m)}{b_i^{m^2}} \right] \quad (4.26)$$

The update equation of the spread parameter b_i^m is in the form:

$$b_i^m(n+1) = b_i^m(n) + m_b [b_i^m(n) - b_i^m(n-1)] - \eta_b \frac{\partial J}{\partial b_i^m} \big|_n \quad (4.27)$$

where

$$\frac{\partial J}{\partial b_i^m} = \frac{\partial J}{\partial J_m} \frac{\partial J_m}{\partial e_p} \frac{\partial e_p}{\partial y_p} \frac{\partial y_p}{\partial \mu_m} \frac{\partial \mu_m}{\partial \mu_{F_i^m}} \frac{\partial \mu_{F_i^m}}{\partial b_i^m} + \dots \quad (4.28)$$

Again for the Gaussian-based function,

$$\frac{\partial \mu_{F_i^m}}{\partial b_i^m} = \mu_{F_i^m} \frac{(x_i^k - c_i^m)^2}{b_i^{m^3}} \quad (4.29)$$

Therefore

$$\frac{\partial J}{\partial b_i^m} = \sum_{k=1}^K \left[\left\{ \sum_{p=1}^P (y_p^k - d_p^k) \frac{L_p^m [y_p^m - y_p^k]}{\sum_{m=1}^M \mu_m^k L_p^m} \right\} \mu_m^k \frac{(x_i^k - c_i^m)^2}{b_i^{m^3}} \right] \quad (4.30)$$

All equations derived are used to update all parameters during the training phase of the network. The initial centre, c_i^m and y_p^m are randomly selected from the k^{th} training data, x_i^k and d_p^k respectively. The initial spread parameter, b_i^m , is determined by

$$b_i = \frac{\max(x_i) - \min(x_i)}{N} \quad (4.31)$$

where b_i is a spread parameter of the i^{th} input of all rules and N is the number of rules. The initial spread parameter, L_p^m , has been set to 0.75 and is adjusted during training.

4.5 Data Analysis

A machine learning approach, based on the AFLS model, has been adopted in order to create a decision support system acting in parallel as an efficient classifier, in an effort to classify meat samples in three quality classes (fresh, semi-fresh, spoiled), as well as a prediction system. The real challenge is to propose a new learning-based structure, which could be considered as a benchmark method towards the development of efficient intelligent methods in food quality analysis. For this reason, produced results are compared against the PLS technique, which is considered as well recognised tool in chemometric analysis. In addition, AFLS's prediction results are compared with those obtained by MLP networks and adaptive neurofuzzy inference system (ANFIS) identification models. Such schemes have become popular modelling techniques in food science and technology in recent years [100]. These approaches are effective and versatile techniques for the identification and modelling of some parameters especially in nonlinear systems. They can be used efficiently to solve problems and to predict parameters in the absence of accurate mathematical models. The final dataset, consisted of 74 beef patterns, include information from the various storage temperatures, the first five PCs and the sampling times.

In this research study, two distinct procedures have been considered. In the first procedure, as the number of observations/samples is small, the separation of the dataset into training and testing subsets (hold-out method) was considered that it would further reduce the number of data and would result in insufficient training of the network. Therefore, in order to improve the robustness of identification process, the Leave-One-Out Cross Validation (LOOCV) technique was employed to evaluate the performance of the developed AFLS model.

The AFLS concept has been developed with the aim to be a more effective modelling tool against to classic MLP and ANFIS structures. Its structure, as shown from Fig. 4.5, consists of an input layer, which in this current research study contains seven input nodes (*i.e.* storage temperature, sampling time, and the values of the five principal components). The second layer is related to the inference engine (*i.e.* the fuzzy rules). After many trials, it has been found that only 12 rules are necessary for the proposed AFLS model to achieve an acceptable performance for this particular case/experiment. The number of membership functions for each input variable is directly associated to the number of rules, hence, each input signal is “distributed” through Gaussian functions with different centres and widths to every rule node via a product operator. The values of the parameters (centres and widths) of the Gaussian membership functions have been adjusted by the learning procedure. The output layer consists of two nodes, corresponding to the predicted quality class (fresh, semi-fresh, spoiled) of meat samples and the total viable counts (TVC), respectively. As both output parameters are dependent, in the sense that quality class is related to microbiological counts and vice

versa, a model that combines both these measurements have been considered to be desirable. In order to accommodate both classification and modelling tasks in the same model-structure, the classification task has been modified accordingly.

Rather than trying to create a distinct classifier, an attempt has been made to “model” the classes [98]. Initially, values of 10, 20 and 30, have been used respectively, to associate the three classes with a cluster centre. During the identification process, output values of [5,15] were associated to “fresh” class with cluster centre 10, values of [15.01,25] were associated to “semi-fresh” class with cluster centre 20, and finally values of [25.01,35] were associated to “spoiled” class with cluster centre 30. The second output node has been assigned to the total viable counts (TVC). The classification accuracy of the AFLS network was determined by the number of correctly classified samples in each sensory class divided by the total number of samples in the class. The performance of the model in the prediction of TVC for each meat sample was determined by the bias (B_f) and accuracy (A_f) factors, the mean relative percentage residual (MRPR) and the mean absolute percentage residual (MAPR), and finally by the root means squared error (RMSE) and the standard error of prediction (SEP) [101].

4.5.1 First Case Study: LOOCV Technique

Results revealed that the classification accuracy of the AFLS model was very satisfactory in the characterisation of beef samples, indicating the advantage of a neurofuzzy approach in tackling complex, nonlinear problems, such as meat spoilage. The classification accuracy obtained from AFLS, is presented in the form of a confusion matrix in Table 4.2.

| True class | Predicted class | | | Row total (n_i) | Sensitivity (%) |
|---|------------------|---------------|------------------|---------------------|-----------------|
| | Fresh | Semi-fresh | Spoiled | | |
| Fresh ($n = 24$) | 23 | 1(marginally) | 0 | 24 | 95.83 |
| Semi-fresh ($n = 16$) | 1 +1(marginally) | 14 | 0 | 16 | 87.5 |
| Spoiled ($n = 34$) | 0 | 0 | 33+1(marginally) | 34 | 100 |
| Column total (n_j) | 25 | 15 | 34 | 74 | |
| Specificity (%) | 92 | 93.33 | 100 | | |
| Overall correct classification (accuracy): 95.94% | | | | | |

Table 4.2 Confusion Matrix for AFLS acting as classifier – LOOCV case

The model overall achieved a 95.94% correct classification, and 95.83%, 87.5% and 100% for fresh, semi-fresh and spoiled meat samples, respectively. The sensitivities (*i.e.* how good the network is at identifying correctly the positive samples) for fresh and spoiled meat samples reveal one (even marginally) out of 24 fresh meat samples, and zero misclassifications out of 34 spoiled samples. One spoiled meat sample was accurately

classified as spoiled, even marginally. In the case of semi-fresh samples, two samples out of 16 were misclassified, as fresh ones. The specificity index (*i.e.* how good the network is at identifying correctly the negative samples) was also high, indicating satisfactory discrimination between these three classes (Table 4.2). It is characteristic that no fresh samples were misclassified as spoiled and vice versa, indicating that the biochemical information provided by FTIR data could discriminate these two classes accurately. Lower percentages were obtained for semi-fresh samples (ca. 87.5%) with incorrect classifications in the fresh class. It must be emphasised however that the number of examined samples within each class was not equally distributed, due to the different spoilage rate of beef samples at the different temperatures. The lower accuracies obtained in the semi-fresh class could be also attributed to the performance of the sensory evaluation process, as the difference between “fresh” and “semi-fresh” class is not very obvious sometimes.

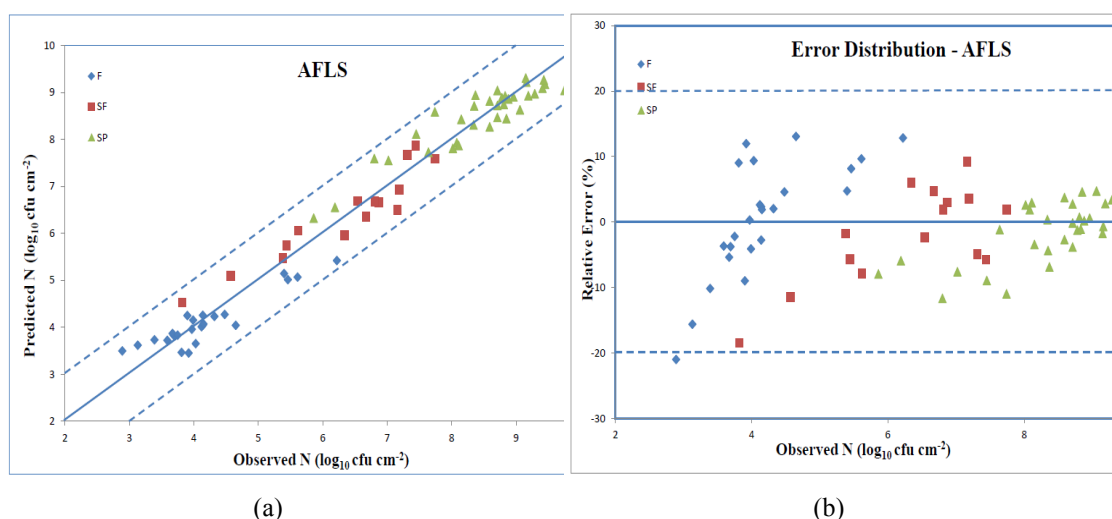


Fig. 4.8 AFLS (a) performance for TVCs, and (b) % relative errors (LOOCV case)

The common measure of goodness-of-fit for model comparison in food microbiology is performed by calculating the squared correlation coefficient (R^2) [102]. The index R^2 ($0 \leq R^2 \leq 1$) is often used as an overall measure of the prediction attained. It measures the fraction of the variation about the mean that is explained by a model. The higher the value, the better is the prediction by the model. AFLS's overall value of 0.98% for the index R^2 indicates a very good fit of the experimental data from the AFLS-based approach. The individual R^2 values for fresh, semi-fresh and spoiled cases were 0.923%, 0.95% and 0.93% respectively. However, R^2 is a criterion for model comparison on the assumption that the error is normally distributed and not dependent on the mean value; In fact, the distribution of the error is not clearly known in the case of microbial/bacteria growth, so this term must be used with caution, particularly in non-linear regression models and hence additional indices must

be employed for model comparison [103]. The plot of predicted versus observed total viable counts is illustrated in Fig. 4.8 (a), and shows a very good distribution around the line of equity ($y=x$), with all the data included within the ± 1 log unit area. However, a few samples were placed near the borderline of the ± 1 log unit area, as shown at Fig. 4.8 (a). These include the spoiled “5F9”, the semi-fresh “0F8” and the fresh “20F6” samples. “5F9” sample corresponds to a beef sample stored at 5°C and collected after 192 h of storage, “0F8” corresponds to a beef sample stored at 0°C and collected after 168 h of storage and finally “20F6” corresponds to a beef sample stored at 20°C and collected after 22 h of storage.

The performance of the AFLS model is also presented in Fig. 4.8 (b), where the % relative error of prediction is illustrated against the observed microbial population. Based on this plot, data was almost equally distributed above and below 0, with all (except one) predicted microbial counts included within the $\pm 20\%$ RE zone. That particular microbial count is associated with the fresh “0F4” sample, which corresponds to a beef fillet, stored at 0°C and collected after 72h of storage. Fig 4.3 reveals the reason for such behaviour. The “0F4” sample is the third sample at the 0°C growth curve. An inspection at the shape of that curve illustrates a deep drop after the first two samples, while the curve is increasing throughout subsequent sampling time. A possible way to overcome this problem could be to broaden the training dataset, especially for low temperatures.

The performance of the AFLS model to predict TVCs in beef samples in terms of statistical indices is presented in Table 4.3. The RMSE values of the AFLS were very low for all samples, with an overall indicator of 0.373. This index is calculated between the desired and output values and then averaged across all data. It can be used as an estimation of the goodness of fit of the models. It can also provide information about how consistent the model would be in the long run [103]. The related RMSE values for the proposed scheme are very low, as shown in Table 4.3, indicating the ability of AFLS to make better prediction on data for which there was no previous training.

Bias factor (B_f) is a multiplicative factor that compares model predictions and is used to determine whether the model over- or under-predicts the response time of bacterial growth. A B_f greater than 1.0 indicates that a growth model is fail-dangerous. Equally, a B_f less than 1.0 generally indicates that a growth model is fail-safe (*i.e.* observed generation times were larger than predicted values), so that predicted values give a margin of safety. Perfect agreement between predictions and observations would lead to B_f of 1. Based on the calculated values of the bias factor B_f , it can be concluded that the AFLS model over-estimated total viable counts in semi-fresh samples ($B_f > 1$), whereas for fresh samples under-estimation of microbial population was evident ($B_f < 1$) and almost perfect for spoiled samples (1.008).

| Statistical index LOOCV (AFLS case) | Mathematical Expression | Fresh | Semi- fresh | Spoiled | Overall |
|---|--|-------|----------------|---------|---------|
| | $P = \text{the predicted values}$ $O = \text{observed values}$ $n = \text{number of observations}$ | | | | |
| Mean squared error (MSE) | $\frac{1}{n} \sum_{i=1}^n (P - O)^2$ | 0.129 | 0.156 | 0.138 | 0.139 |
| Root mean squared error (RMSE) | $\sqrt{\frac{\sum (O - P)^2}{n}}$ | 0.359 | 0.395 | 0.371 | 0.373 |
| Mean relative percentage residual (MRPR %) | $\frac{100}{n} \times \sum \frac{(O - P)}{O}$ | 0.648 | -2.431 | -0.964 | -0.758 |
| Mean absolute percentage residual (MAPR %) | $\frac{100}{n} \times \sum \frac{ O - P }{O}$ | 7.091 | 6.234 | 3.726 | 5.359 |
| Bias factor (B_f) | $10^{\sum \log(P/O)/n}$ | 0.989 | 1.022 | 1.008 | 1.005 |
| Accuracy factor (A_f) | $10^{\sum \log(P/O)/n}$ | 1.074 | 1.063 | 1.037 | 1.054 |
| Standard error of prediction (SEP %) | $\frac{100}{O} \sqrt{\frac{\sum (O - P)^2}{n}}$ | 8.579 | 6.356 | 4.406 | 5.671 |

Table 4.3 Performance of AFLS - LOOCV case

The accuracy factor A_f is a simple multiplicative factor that indicates the spread of results about the prediction. A value of one indicates that there is perfect agreement between all the predicted and measured values. In our case, the values of the accuracy factor A_f indicated that the predicted total viable counts were 7.37%, 6.26%, and 3.74% different from the observed values for fresh, semi-fresh, and spoiled meat samples, respectively. Regarding the appropriate values of the accuracy factor A_f , it has been reported [103] that an increase of 0.15 (15%) would be acceptable for each independent variable included in model development. Therefore, in our study, with only one independent variable (*i.e.* temperature), any value of A_f up to 1.15, could be considered to be satisfactory. The mean relative percentage residual index (MRPR) similarly verified the over-prediction for semi-fresh and spoiled samples (MRPR < 0) and under-prediction for fresh samples (MRPR > 0), whereas the values of mean absolute percentage residual (MAPR), representing the average deviation between observed and predicted counts, verified the information provided by the accuracy factor. Finally, the standard error of prediction (SEP) index is a relative typical deviation of the mean prediction values and expresses the expected average error associated with future predictions. It has the advantage of being independent on the magnitude of the measurements [103]. The lower the value of this index is, the better the capability of the model to predict microbial counts in new meat samples. The value of the index was 5.67% for the overall samples indicating good performance of the network for microbial count predictions in this

class (Table 4.3), with also very low values (4.4%) for spoiled samples. However in the case of fresh samples, the index gave higher values (*i.e.* 8.58%) as the network under-estimated microbial counts for some fresh beef samples.

Although AFLS identification model utilises the gradient descent learning method for training, its main advantage is related to its MIMO structure capability. The majority of existing neurofuzzy schemes follow the classic Takagi–Sugeno–Kang (TSK) structure, where only one output is enabled. TSK models consist of IF-THEN rules with fuzzy antecedents and mathematical functions in the consequent part. The fuzzy sets partition the input space into a number of fuzzy regions, while the consequent functions describe the system's behaviour in these regions. ANFIS is a classic representative of TSK-based neurofuzzy systems [104]. By analysing mapping relationships between input and output data, ANFIS optimises the distribution of membership functions by using a gradient descent algorithm either alone or combined with a least-squares method. The ANFIS uses fuzzy if-then rules involving premise and consequent parts of an TSK-type fuzzy inference system. The five-layer system ANFIS architecture includes a fuzzification layer (Layer 1), a production layer (Layer 2), a normalisation layer (Layer 3), a defuzzification layer (Layer 4), and a total output layer (Layer 5) [105].

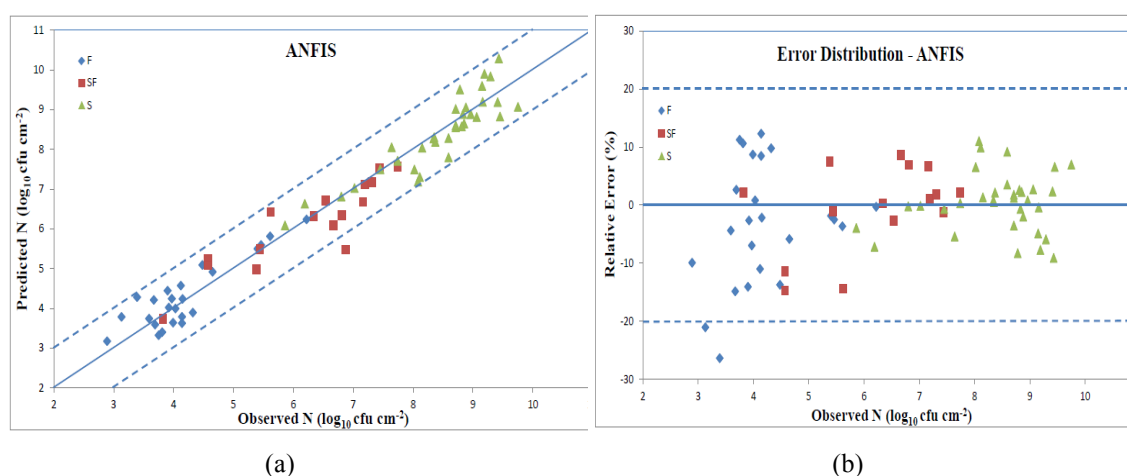


Fig. 4.9 ANFIS (a) performance for TVCs, and (b) % relative errors (LOOCV case)

In addition to AFLS, in this research work, an ANFIS model has been implemented to predict TVCs. The same leave-one-out cross validation technique, as well as the same training dataset have been utilised also for this case. Under these conditions, ANFIS performed very satisfactory, its performance however was achieved with a high computational cost, by utilising two membership functions for each input variables and 128 fuzzy rules. The related plot of the predicted versus the observed TVCs, as shown in Fig. 4.9 (a), reveals a good distribution around the line of equity, with the majority of data (*ca.* 98.65%) included within the ± 1 log unit area. Four samples were in the borderline of the ± 1 log unit area, while one

sample (semi-fresh) was placed outside that unit area. More specifically, the semi-fresh “10F7” sample was clearly outside the ± 1 log unit area. “10F7” corresponds to a beef sample stored at 10°C and collected after 52 h of storage. Spoiled samples, “10F12”, “20F11” and fresh samples “0F6”, “0F5” were placed very close to the borderline. “10F12” sample corresponds to a beef sample stored at 10°C and collected after 90 h of storage, while “20F11” corresponds to a beef sample stored at 20°C and collected after 44 h of storage. Similarly, 0F6” sample corresponds to a beef sample stored at 0°C and collected after 120 h, while “0F5” corresponds to a beef sample stored at 0°C and collected after 96 h of storage.

| Statistical index LOOCV (ANFIS case) | Fresh | Semi-fresh | Spoiled | Overall |
|---|---------|------------|---------|---------|
| <i>Mean squared error (MSE)</i> | 0.1580 | 0.2717 | 0.1881 | 0.1964 |
| <i>Root mean squared error (RMSE)</i> | 0.3975 | 0.5213 | 0.4337 | 0.4432 |
| <i>Mean relative percentage residual (MRPR %)</i> | -3.1892 | 0.7611 | 0.3848 | -0.6930 |
| <i>Mean absolute percentage residual (MAPR %)</i> | 8.5735 | 6.4222 | 3.9022 | 5.9621 |
| <i>Bias factor (B_f)</i> | 1.0269 | 0.9886 | 0.9949 | 1.0038 |
| <i>Accuracy factor (A_f)</i> | 1.0868 | 1.0670 | 1.0400 | 1.0608 |
| <i>Standard error of prediction (SEP %)</i> | 9.4997 | 8.3852 | 5.1477 | 6.7435 |

Table 4.4 Performance of ANFIS - LOOCV case

The performance of the ANFIS model is also presented in Fig. 4.9 (b), where the % relative error of prediction is illustrated against the observed microbial population. Based on this plot, data was almost equally distributed above and below 0, with all (except two) predicted microbial counts included within the $\pm 20\%$ RE zone. These particular microbial counts are associated with the fresh “0F5” and “0F6” samples. The performance of the ANFIS model in predicting TVC in meat samples in terms of statistical indices is presented in Table 4.4.

| True class | Predicted class | | | Row total (n_i) | Sensitivity (%) |
|--|-----------------|------------------|------------------|---------------------|-----------------|
| | Fresh | Semi-fresh | Spoiled | | |
| Fresh ($n = 24$) | 21 | 1(marginally) +2 | 0 | 24 | 87.5 |
| Semi-fresh ($n = 16$) | 2 | 13 | 1 | 16 | 81.25 |
| Spoiled ($n = 34$) | 0 | 1 | 31+2(marginally) | 34 | 97.06 |
| Column total (n_j) | 23 | 17 | 34 | 74 | |
| Specificity (%) | 91.30 | 76.47 | 97.06 | | |
| Overall correct classification (accuracy): 90.54% | | | | | |

Table 4.5 Confusion Matrix for MLP acting as classifier – LOOCV case

An MLP network has been also implemented using the same FTIR dataset and employing the leave-one-out validation method. After a few trials, the MLP was constructed with two

hidden layers (with 12 and 6 nodes respectively) and two output nodes, one for the sensory class and one for the TVCs. The classification accuracy obtained from MLP, is presented in the form of a confusion matrix in Table 4.5. The model overall achieved a 90.54% correct classification, with 87.5%, 81.25% and 97.06% for fresh, semi-fresh and spoiled meat samples, respectively. The related sensitivities represent 3 misclassifications out of 24 fresh meat samples, and one misclassification out of 34 spoiled samples. In the case of semi-fresh samples the respective figure was somehow lower (81.25%). In this case, 3 samples were misclassified (out of 16), two as fresh and one as spoiled cases.

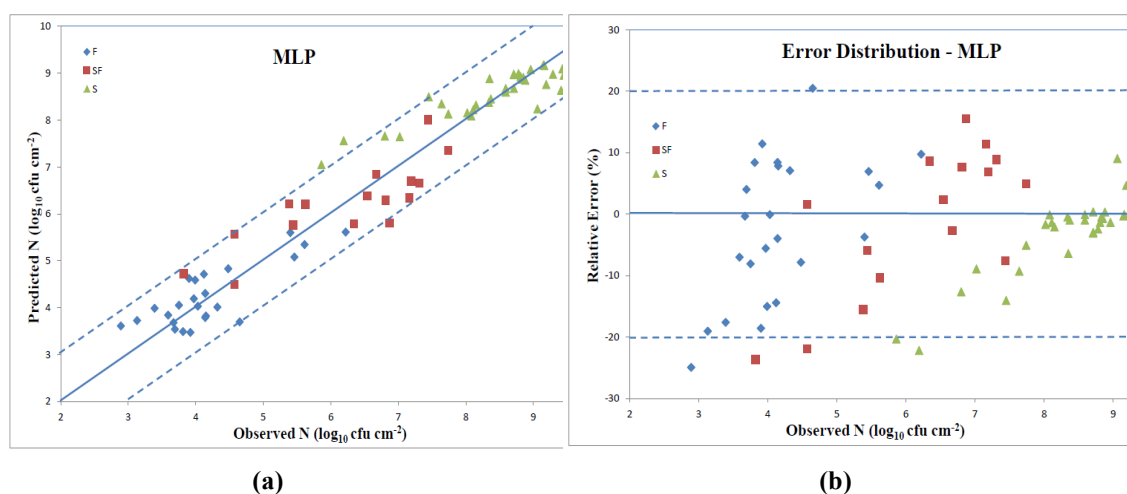


Fig. 4.10 MLP (a) performance for TVCs, and (b) % relative errors (LOOCV case)

The related plot of the predicted versus the observed TVCs, as shown in Fig. 4.10 (a), reveals a reasonable good distribution around the line of equity without any particular trend, with the majority of data (*ca.* 90.54%) included within the ± 1 log unit area. Three samples were in the borderline of the ± 1 log unit area, while four samples (3 spoiled and one semi-fresh) were placed outside that unit area. A more comprehensive picture of the prediction performance of the MLP is given in Fig. 4.10 (b) where the % relative error of prediction is depicted against the observed microbial population. Based on this plot, approximately 91.89% of predicted microbial counts included within the $\pm 20\%$ RE zone.

The performance of the MLP network in predicting TVC in meat samples in terms of statistical indices is presented in Table 4.6. An MLP implementation of the same FTIR dataset has been considered recently, however with a much simpler structure [81]. More specifically, an MLP with one hidden layer (with 10 nodes) and two outputs managed to perform the same task, however with worse results. The overall RMSE was 0.850, while the overall SEP was increased to 12.94%. It is well known that a two-hidden MLP structure enjoys a better performance against its single-hidden MLP counterpart, and such results simply proved that concept.

| Statistical index LOOCV (MLP case) | Fresh | Semi-fresh | Spoiled | Overall |
|---|---------|------------|---------|---------|
| <i>Mean squared error (MSE)</i> | 0.2082 | 0.4077 | 0.2856 | 0.2869 |
| <i>Root mean squared error (RMSE)</i> | 0.4563 | 0.6385 | 0.5344 | 0.5357 |
| <i>Mean relative percentage residual (MRPR %)</i> | -2.3964 | -1.2590 | -2.1948 | -2.0578 |
| <i>Mean absolute percentage residual (MAPR %)</i> | 9.8067 | 9.7132 | 4.8452 | 7.5069 |
| <i>Bias factor (B_f)</i> | 1.0175 | 1.0062 | 1.0196 | 1.0160 |
| <i>Accuracy factor (A_f)</i> | 1.1008 | 1.0999 | 1.0480 | 1.0760 |
| <i>Standard error of prediction (SEP %)</i> | 10.9063 | 10.2712 | 6.3441 | 8.1509 |

Table 4.6 Performance of MLP - LOOCV case

Although both AFLS and MLP share the same learning training algorithm, *i.e.* the gradient descent method, the different “philosophy” in building the neurofuzzy architecture, allowed AFLS to achieve such superior performance. In MLP models, all normalised inputs are fed to the hidden layer, while in the case of AFLS, each input is fuzzified / decomposed through Gaussians membership functions. As the number of these membership functions is equal to the numbers of rules, this architecture has advantages over the classic ANFIS neurofuzzy model. The increased number of Gaussian membership functions increases the localisation of the input signal while at the same time maintains the required number of rules at low level. This localisation spread through the membership functions, is one advantage against the classic MLP structure. The proposed defuzzification scheme improves also the final output, against a simple sigmoid function, as it is the case for MLP. All modelling schemes have been implemented in MATLAB (ver. R2012a, Mathworks.com).

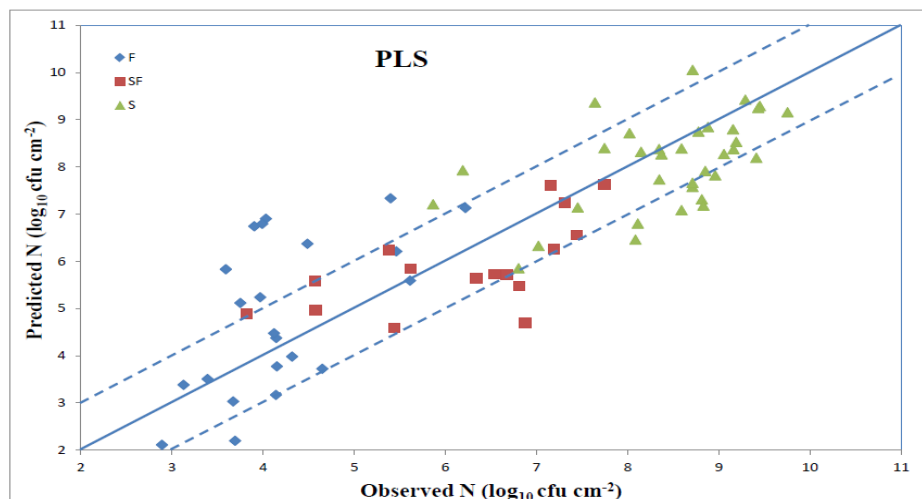


Fig 4.11 PLS performance for TVCs

Partial least squares (PLS) regression approaches, similarly to NNs, have the ability to relate the input and output variables without having any pre-knowledge on physics of the system

provided an accurate and adequate amount of data on the system variables is available [105]. The PLS method is a linear multivariate regression method that projects the input–output data down in to a latent space, extracting a number of principal factors with an orthogonal structure, while capturing most of the variance in the original data. PLS derives its usefulness from its ability to analyse data with strongly collinear, noisy and numerous variables in the predictor matrix X and response Y . The PLS model was constructed using the same input vector as in the cases of AFLS, ANFIS and MLP, and the PLS_Toolbox software (ver. 7.5, Eigenvector.com) in association with MATLAB was used to perform the PLS analysis. The nonlinear iterative partial least squares algorithm (NIPALS) has been chosen as the appropriate learning scheme. The related plot of the predicted versus the observed TVCs, as shown in Fig. 4.11, reveals a satisfactory, but inferior to MLP and AFLS, distribution around the line of equity without any particular trend, with the majority of data (*ca.* 67.56%) included within the ± 1 log unit area. The performance of the PLS regression model in predicting TVC in meat samples in terms of statistical indices is presented in Table 4.7.

| Statistical index LOOCV (PLS case) | Fresh | Semi-fresh | Spoiled | Overall |
|---|---------|------------|---------|---------|
| <i>Mean squared error (MSE)</i> | 2.6621 | 0.8929 | 0.9514 | 1.4936 |
| <i>Root mean squared error (RMSE)</i> | 1.6316 | 0.9449 | 0.9754 | 1.2221 |
| <i>Mean relative percentage residual (MRPR %)</i> | -7.6389 | 2.9306 | 3.4914 | -0.2397 |
| <i>Mean absolute percentage residual (MAPR %)</i> | 32.0962 | 13.5635 | 9.9630 | 17.9199 |
| <i>Bias factor (B_p)</i> | 0.9666 | 0.9581 | 0.9582 | 0.9609 |
| <i>Accuracy factor (A_p)</i> | 1.4270 | 1.1503 | 1.1071 | 1.2121 |
| <i>Standard error of prediction (SEP %)</i> | 38.9944 | 15.2013 | 11.5779 | 18.5961 |

Table 4.7 Performance of PLS - LOOCV case

Although in general, PLS results are worse than those from MLP, special attention should be paid to the cases of fresh samples. Both SEP and Accuracy Factor indicators are very extremely high and such behaviour could be explained by an inspection at Fig. 4.11. Four fresh samples, “15F2”, “20F4”, “20F2” and “20F3”, which correspond to collection after 6 h, 12 h, 4 h and 8 h of storage respectively, are placed far away from the ± 1 log unit area. Fig. 4.3 illustrates the behaviour of 15°C and 20°C growth curves, where a very abrupt behaviour can be noticed especially for the 20°C curve at low sampling times. However, such results from PLS are expected, as it is well known that in modelling of real processes, linear PLS has some difficulties in its practical applications since most real problems are inherently nonlinear and dynamic [106].

4.5.2 Second Case Study: Hold-Out Method

The FTIR data analysis utilising the leave-one-out cross validation method was imposed by the relative number of experimental samples. In order to investigate further the capabilities of AFLS model in this joint problem of classification/prediction, a second experiment was carried out, where the initial FTIR dataset was divided into a training subset with approx. 75% of the data, and a testing subset with the remaining 25% (*i.e.* 19 samples). An ANFIS model and an MLP network have been also developed for comparison reasons, to associate the same spectral data from FTIR analysis with beef fillet spoilage during aerobic storage at different temperatures.

| True class | Predicted class | | | Row total (n_i) | Sensitivity (%) |
|---|----------------------|------------|---------|---------------------|-----------------|
| | Fresh | Semi-fresh | Spoiled | | |
| Fresh ($n = 7$) | 6+1(very marginally) | 0 | 0 | 7 | 100 |
| Semi-fresh ($n = 5$) | 0 | 4 | 1 | 5 | 80 |
| Spoiled ($n = 7$) | 0 | 0 | 7 | 7 | 100 |
| Column total (n_j) | 7 | 4 | 8 | 19 | |
| Specificity (%) | 100 | 100 | 87.5 | | |
| Overall correct classification (accuracy): 94.74% | | | | | |

Table 4.8 Confusion Matrix for AFLS acting as classifier – Case 2

The proposed AFLS model has been also utilised in this second simulation study, in order to assess its competence to be trained with a dataset with a reduced number of samples. For this particular case, after trials, it has been found that 15 rules were necessary for the proposed model to achieve an acceptable performance for this particular case/experiment. The training set consisted of 55 samples, while 19 (7 fresh, 5 semi-fresh and 7 spoiled) meat samples were included in the testing subset. Table 4.8 illustrates these testing results. It has to be mentioned however, that one fresh meat sample, although formally categorised as positive, its value was almost identical to the cut-off threshold, therefore it could be considered it as a “grey” case. In addition, AFLS’s second output modelled successfully the TVCs, as illustrated in Fig. 4.12. In this case, the plot of the predicted vs. the observed TVC for the testing dataset, have shown reasonably good distribution around the line of equity without any particular trend, with the majority of data (*ca.* 89.47%) included within the ± 1 log unit area. The semi-fresh “10F7” sample is clearly out of the ± 1 log unit area, while the spoiled “5F9” sample is on the borderline. Another fresh sample (*i.e.* “0F5”) has been placed very close to the borderline. “10F7” sample corresponds to a beef sample stored at 10°C and collected after 52 h of storage, “5F9” corresponds to a beef sample stored at 5°C and collected after 192 h of storage

and finally “0F5” corresponds to a beef sample stored at 0°C and collected after 96 h of storage.

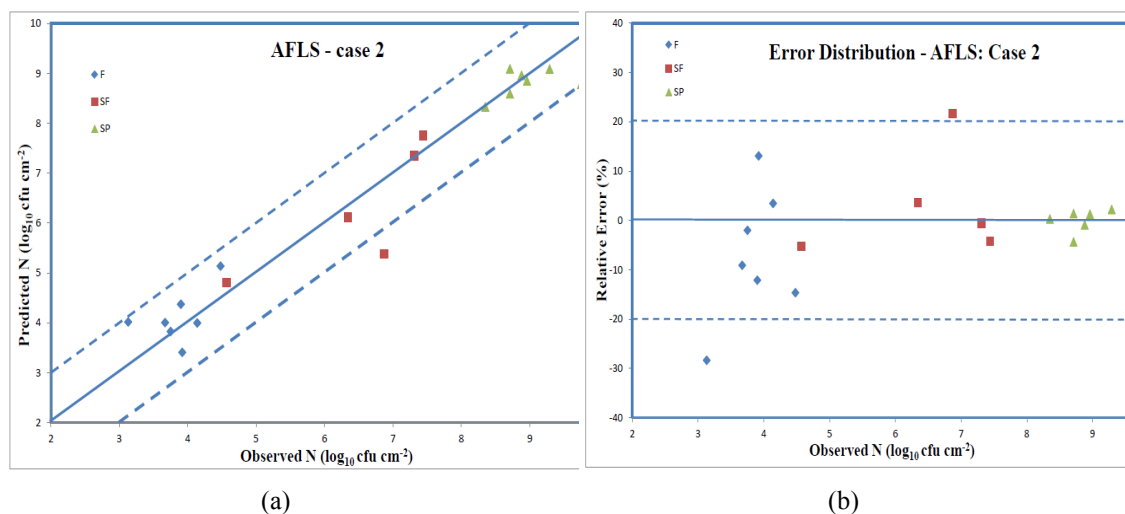


Fig. 4.12 AFLS (a) performance for TVCs, and (b) % relative errors (Case 2)

The comparison of Fig. 4.12a with the related Fig. 4.8a is more than evident. A more comprehensible picture of the AFLS’s prediction performance is provided in Fig. 4.12b, where the % relative error of prediction is shown against the observed microbial population. Based on this plot, data were distributed above and below 0, with approximately 89.47% of predicted microbial counts included within the $\pm 20\%$ RE zone. Samples “10F7” and “0F5” are placed outside the $\pm 20\%$ RE zone.

| Statistical index - AFLS (19 test) | Fresh | Semi-fresh | Spoiled | Overall |
|---|--------|------------|---------|---------|
| <i>Mean squared error (MSE)</i> | 0.263 | 0.484 | 0.166 | 0.286 |
| <i>Root mean squared error (RMSE)</i> | 0.513 | 0.696 | 0.407 | 0.534 |
| <i>Mean relative percentage residual (MRPR %)</i> | -7.107 | 3.052 | 1.404 | -1.298 |
| <i>Mean absolute percentage residual (MAPR %)</i> | 11.828 | 7.045 | 2.906 | 7.282 |
| <i>Bias factor (B_f)</i> | 1.064 | 0.964 | 0.985 | 1.007 |
| <i>Accuracy factor (A_f)</i> | 1.118 | 1.078 | 1.031 | 1.075 |
| <i>Standard error of prediction (SEP %)</i> | 13.311 | 10.695 | 4.552 | 8.311 |

Table 4.9 Performance for AFLS - Case 2

The performance of the AFLS model to predict TVCs in beef samples for this second simulation, in terms of statistical indices is presented in Table 4.9. Based on the calculated values of the bias factor B_f , it can be assumed that the neurofuzzy network under-estimated TVCs in semi-fresh and spoiled samples ($B_f < 1$), whereas for fresh samples over-estimation

of microbial population was evident ($B_f > 1$). The overall B_f was almost optimal (*ca.* 1.007). In addition, the values of the accuracy factor A_f indicated that the predicted TVCs were 11.8%, 7.8%, and 3.1% different from the observed values for fresh, semi-fresh, and spoiled meat samples, respectively. However, a closer comparison of AFLS performance for these two simulation case studies reveals a problem with the limited number of samples for training. The SEP index is much worse in this second case, and this reflects an open problem in learning-based systems, *i.e.* the need to have as large as possible training datasets. Similarly to the previous case study, an ANFIS model has been developed to predict TVCs for this reduced dataset. ANFIS's performance although generally inferior to the previous case study, revealed an almost excellent “response” to the cases of semi-fresh and spoiled samples. Unfortunately, such performance was compromised with a high computational cost, by utilising two membership functions for each input variable and 128 fuzzy rules.

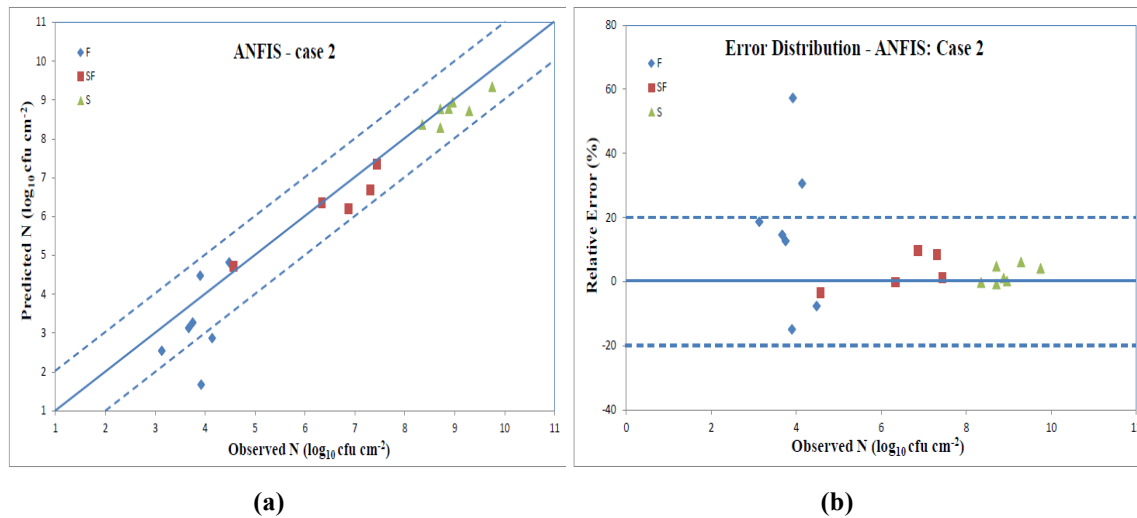


Fig. 4.13 ANFIS (a) performance for TVCs, and (b) % relative errors (Case 2)

The related plot of the predicted versus the observed TVCs, as shown in Fig. 4.13a, reveals a good distribution around the line of equity without any particular trend, with the majority of data (*ca.* 89.47%) included within the ± 1 log unit area. Two fresh samples were clearly outside the borderline of the ± 1 log unit area. A more complete picture of the prediction performance of the ANFIS model is given in Fig. 4.13b where the % relative error of prediction is depicted against the observed microbial population.

These diagrams reveal however some similarities. Samples “0F2” and “10F3” which are placed outside the ± 1 log unit area at Fig. 4.13a also are responsible for the high relative errors at Fig. 4.13b, with 57.32% and 30.63% respectively. Sample “0F2” corresponds to a beef sample stored at 0°C and collected after 24 h of storage, while sample “10F3” to a beef sample stored at 10°C and collected after 18 h of storage.

The performance of the ANFIS model in predicting TVCs for beef samples for this second simulation, in terms of statistical indices is presented in Table 4.10.

| Statistical index - ANFIS (19 test) | Fresh | Semi-fresh | Spoiled | Overall-test |
|---|---------|------------|---------|--------------|
| <i>Mean squared error (MSE)</i> | 1.1384 | 0.1699 | 0.0970 | 0.4998 |
| <i>Root mean squared error (RMSE)</i> | 1.0670 | 0.4122 | 0.3115 | 0.7070 |
| <i>Mean relative percentage residual (MRPR %)</i> | 15.9330 | 3.1366 | 2.1999 | 7.5060 |
| <i>Mean absolute percentage residual (MAPR %)</i> | 22.3553 | 4.5608 | 2.4782 | 10.3494 |
| <i>Bias factor (B_p)</i> | 0.8064 | 0.9673 | 0.9777 | 0.9081 |
| <i>Accuracy factor (A_p)</i> | 1.3176 | 1.0484 | 1.0257 | 1.1313 |
| <i>Standard error of prediction (SEP %)</i> | 27.6720 | 6.3350 | 3.4800 | 10.9953 |

Table 4.10 Performance for ANFIS - Case 2

The performance of the MLP model using the reduced training dataset, although it could be considered as satisfactory, it is also inferior compared to MLP's performance from the previous case. The calculated correct classifications were similar to the previous case (*i.e.* leave-one-out case). More specifically, the classification for fresh, semi-fresh and spoiled beef samples were 85.71%, 80.0% and 100% respectively, whereas the overall correct classification (accuracy) for MLP test dataset was 89.47%. The semi-fresh "15F10" sample, which corresponds to a beef sample stored at 15°C and collected after 48 h of storage, was classified as spoiled, whereas the fresh "15F5" sample, which corresponds to a beef sample stored at 15°C and collected after 24 h of storage, was classified as semi-fresh case. Table 4.11 illustrates the MLP-based testing classification results.

| True class | Predicted class | | | Row total (n_i) | Sensitivity (%) |
|--|-----------------|------------|---------|---------------------|-----------------|
| | Fresh | Semi-fresh | Spoiled | | |
| Fresh ($n = 7$) | 6 | 1 | 0 | 7 | 85.71 |
| Semi-fresh ($n = 5$) | 0 | 4 | 1 | 5 | 80.0 |
| Spoiled ($n = 7$) | 0 | 0 | 7 | 7 | 100 |
| Column total (n_j) | 6 | 5 | 8 | 19 | |
| Specificity (%) | 100 | 80 | 87.50 | | |
| Overall correct classification (accuracy): 89.47% | | | | | |

Table 4.11 Confusion Matrix for MLP acting as classifier – Case 2

The related plot of the predicted versus the observed TVCs, as shown in Fig. 4.14a, reveals a reasonable good distribution around the line of equity without any particular trend, with the majority of data (*ca.* 84.21%) included within the ± 1 log unit area. Three samples (2 fresh and one spoiled) were outside the borderline of the ± 1 log unit area. A more complete picture

of the prediction performance of the MLP is given in Fig. 4.14b where the % relative error of prediction is depicted against the observed microbial population. Based on this plot, approximately 78.94% of predicted microbial counts included within the $\pm 20\%$ RE zone.

The below two diagrams reveal also some resemblances. Samples “15F5” and “0F2” which are placed outside the ± 1 log unit area at Fig. 4.14a also are responsible for the high relative errors at Fig. 4.14b, with -48.76% and 28.86% respectively. Sample “0F2” corresponds to a beef sample stored at 0°C and collected after 24 h of storage. The performance of the MLP model in predicting TVCs for beef samples for this second simulation, in terms of statistical indices is presented in Table 4.12.

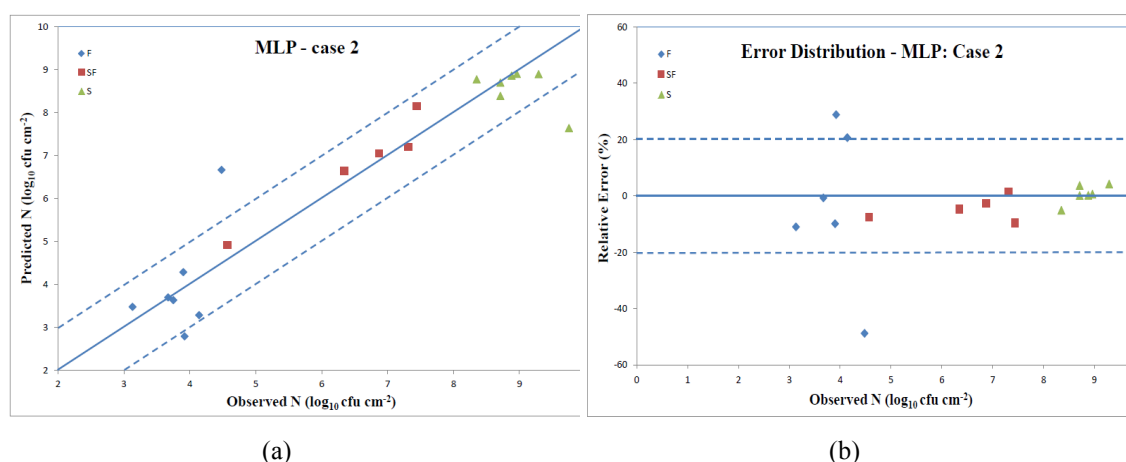


Fig. 4.14 MLP (a) performance for TVCs, and (b) % relative errors (Case 2)

The overall RMSE for the MLP was 0.8133, while the overall SEP was increased to 12.73%. However, an interesting issue from both Fig. 4.14a and Table 4.12 is related with the performance of semi-fresh samples. In comparison with the related AFLS and ANFIS cases, MLP model managed to predict more accurately the growth dynamics of semi-fresh samples. The related SEP was 5.97%, compared to 10.7% and 6.33% for the cases of AFLS and ANFIS respectively.

| Statistical index - MLP (19 test) | Fresh | Semi-fresh | Spoiled | Overall-test |
|---|---------|------------|---------|--------------|
| <i>Mean squared error (MSE)</i> | 1.0099 | 0.1509 | 0.7000 | 0.6697 |
| <i>Root mean squared error (RMSE)</i> | 1.0050 | 0.3884 | 0.8367 | 0.8183 |
| <i>Mean relative percentage residual (MRPR %)</i> | -2.4966 | -4.5711 | 3.6519 | -0.7773 |
| <i>Mean absolute percentage residual (MAPR %)</i> | 17.5794 | 5.1790 | 5.1044 | 9.7201 |
| <i>Bias factor (B_f)</i> | 0.9995 | 1.0450 | 0.9600 | 0.9964 |
| <i>Accuracy factor (A_f)</i> | 1.1883 | 1.0514 | 1.0566 | 1.1019 |
| <i>Standard error of prediction (SEP %)</i> | 26.0641 | 5.9699 | 9.3482 | 12.7269 |

Table 4.12 Performance for MLP - Case 2

Overall results revealed that both classification and especially prediction accuracies of the AFLS model were better compared with the performances of MLP and ANFIS in the characterisation of meat samples for this reduced number of samples, indicating again the superiority of this specific MIMO neurofuzzy approach in tackling complex, nonlinear problems such as the meat spoilage.

In this chapter, Fourier transform infrared spectra were used to identify meat spoilage microorganisms during aerobic storage at chill and abuse temperatures. The identification strategy involved not only the classification of beef samples in their respective quality class (*i.e.* fresh, semi-fresh and spoiled), but also the prediction of their associated microbiological population directly from FTIR spectra. The realisation of this strategy has been fulfilled with the development of a MIMO neurofuzzy model which incorporates a prototype defuzzification scheme, while utilising an efficient, compared to TSK-systems, fuzzification layer. In the case of AFLS, the number of memberships for each input variable was directly associated to the number of rules, hence, the “*curse of dimensionality*” problem was significantly reduced. Classification performance was almost excellent, with 95.94% and 94.74% accuracy for the two different case studies. Similarly, overall prediction for TVCs has been considered as very satisfactory, although lower performance was observed especially for the fresh samples. ANFIS’s prediction performance appeared to be comparable to AFLS case; however such results were achieved with huge expensive computational cost. ANFIS suffers from the “*curse of dimensionality*” problem as well as the inability to support multiple output variables. Prediction performances of MLP and PLS schemes revealed the deficiencies of these systems which have been used extensively in the area of Food Microbiology.

Chapter Five

COMPUTATIONAL INTELLIGENCE TECHNIQUES TO ASSESS SENSORY QUALITY OF MEAT USING MULTISPECTRAL IMAGING

5.1 Introduction

Machine vision and Near Infrared (NIR) spectroscopy are two of the most extensively applied methods for food quality and safety assessment. The use of vision technology for quality testing of food production has the obvious advantage of being able to continuously monitor a production using non-destructive methods, thus increasing the quality and minimizing cost. Machine vision techniques based on RGB colour vision systems have been successfully applied to evaluate the external characteristics of foods [108]. Such systems, however, are not able to capture broad spectral information, which is related to internal characteristics; hence computer vision has limited ability to conduct quantitative analysis of chemical components in food [109]. Alternatively, the tight relationship between NIR spectra and food components makes NIR spectroscopy more attractive than the other spectroscopic techniques. On the other hand, these spectral methods were proved inefficient when it comes to heterogeneous materials such as meat, owing to the fact that they are not capable of obtaining any spatial information about objects [110].

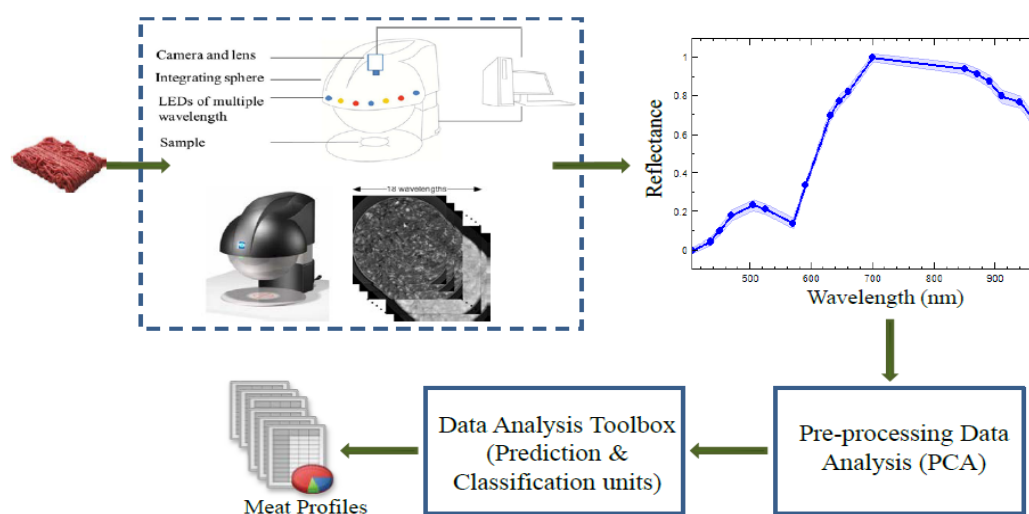


Fig. 5.1 Schematic of the proposed data analysis

In recent years, spectral imaging (i.e., hyperspectral and multispectral) has emerged as a better tool for safety and quality inspection of various agricultural commodities. This technique integrates the conventional imaging and spectroscopy technique to attain both spatial and spectral information from the target objects simultaneously.

The main objective of this chapter is to associate, for the first time according to literature, spectral data acquired by multispectral imaging techniques with meat spoilage, using neurofuzzy systems. Fig. 5.1 illustrates the proposed data analysis concept.

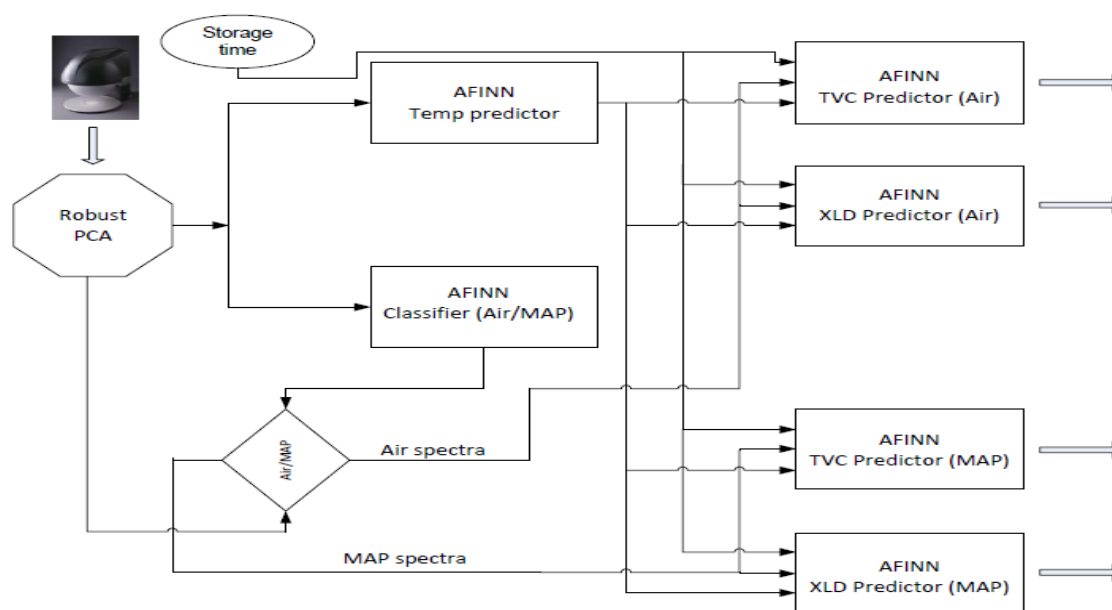


Fig.5.2 Structure of proposed decision support system

Minced beef samples, packaged either aerobically (AIR) or under modified atmosphere (MAP), were held from freshness to spoilage at 0, 5, 10, and 15°C. Datasets related to imaging spectral Agricultural University of Athens, Greece, again provided information and the associated microbiological analysis from meat samples. An intelligent decision support system has been designed in such way in order to accommodate all relevant information. Its overall schematic diagram shown at Fig. 5.2 includes a classifier unit to discriminate AIR/MAP based samples as well as an identification model to predict the temperatures under which meat samples were stored.

The proposed approach is considered as novel, as for the first time, prediction of temperature is performed utilizing only imaging spectral information. Individual identification models have been also developed for the prediction of the total viable counts of bacteria (TVC) as well as the growth of salmonella (XLD) for both AIR/MAP conditions. The Adaptive Fuzzy Inference Neural Network (AFINN), a Takagi–Sugeno–Kang (TSK) structure, has been considered as the identification/classifier models for this proposed decision support systems [111]. Results from AFINN scheme are compared against models based on ANFIS, multilayer neural networks (MLP) and PLS schemes.

5.2 Multispectral Imaging Approaches in Food Quality

Analysis

Due to its simplicity, hyperspectral / multispectral imaging has been utilised for online process monitoring and quality control applications [112]. Recently, this technique has been adopted to assess food safety and quality including contaminants detection [113]. In addition, inspection of internal and external attributes in various fruits and vegetables such as apple [114] and fresh-cut spinach leaves [115] have been performed using multispectral imaging combined with various chemometric methods. Hyperspectral and multispectral imaging spectroscopy have been also used as rapid techniques to monitor quality attributes of meat products. A non-destructive method based on multispectral imaging in the visible and near infrared (NIR) regions to determine the aerobic plate count (APC) in cooked pork sausages has been considered recently [116]. The prediction of total viable counts of minced pork meat stored under two different storage conditions - aerobic and modified atmosphere packages - has been performed using the VideometerLab multispectral imaging device [117]. A hyperspectral imaging technique has been investigated for evaluating pork meat tenderness and *Escherichia coli* contamination [118]. In that research study, a Modified Gompertz function was exploited to extract the scattering characteristics of pork meat from the spatially-resolved hyperspectral images. Hyperspectral images were used to predict fresh beef tenderness based on Warner–Bratzler shear force [119]. The identification and extraction of useful colour and texture features from fresh beef samples using a multispectral imaging system has been also explored and a support vector machine algorithm was then utilised to predict cooked beef tenderness [120]. The detection of minced lamb adulteration has been considered using hyperspectral imaging [121], while a feasibility of combining spectral with texture features in order to improve pH prediction for salted pork was investigated through hyperspectral imaging [122]. Finally, detection of adulteration of minced beef with pork samples has been explored using multispectral image analysis [123].

5.3 Multispectral Imaging - Sampling and Analysis

The entire experimental case study was performed at the Agricultural University of Athens, Greece. Minced Meat was separated into small portions (75 g) and packaged individually either aerobically or under modified atmosphere (MAP) (40% CO₂, 30% O₂, 30% N₂), and in different temperatures (0, 5, 10, 15 °C) that are associated with acceptable/non-acceptable storage practices in a distribution chain for meat products [124]. Fig 5.3 shows a sample of meat under these different storage conditions.

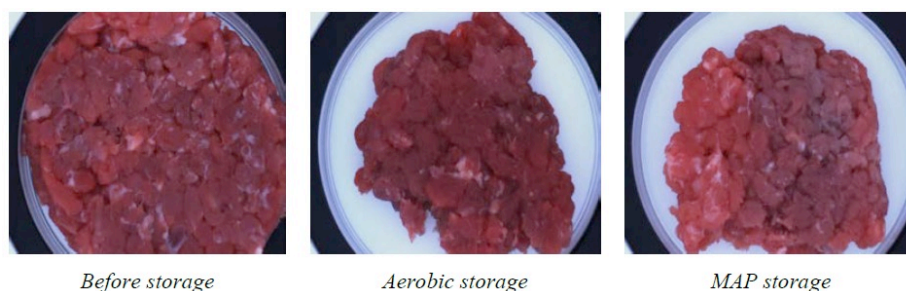


Fig. 5.3 Example of Aerobic vs. MAP storage

At the beginning and during storage, after appropriate time intervals, meat samples were divided into two parts; one part was used for microbiological analysis while the other one for image analysis. It was assumed that the microbial population at both parts would be comparable. Aerobic samples stored at 0°C and 5°C were analysed approx. every 48h for the period $0-186\text{h}$ and every 24h for the period $186-378\text{h}$. Finally, the last sample was analysed at 479h . Similarly, samples stored at 10°C and 15°C were analysed approx. every 12h for the period $0-156\text{h}$. In total, 14 samples were analysed for each temperature case, resulting 56 samples in total. The same procedure was repeated for MAP case too. Microbiological analysis was performed, and resulting growth data from plate counts were \log_{10} transformed and fitted to the Baranyi & Roberts' model in order to verify the kinetic parameters of microbial growth (maximum specific growth rate and lag phase duration) for the TVC and salmonella (XLD). A detailed description of the preparation of minced beef samples, as well as their related microbiological analysis, is described in [124]. The growth curves of TVC and XLD for minced beef storage at different temperatures under AIR and MAP conditions as a function of storage time are illustrated in Fig 5.4.

The growth curves for both TVC cases are similar, with the exception that the maximum specific growth rate (μ_{max}) for the AIR packaged condition is different than of that of the MAP case. It has been found that packaging under modified atmosphere delay the growth rates of all members of the microbial association, as well as the maximum population attained by each microbial group compared with aerobic storage. Aerobic storage accelerates spoilage due to the fast growing *Pseudomonas* spp.; in addition such growth can be significantly inhibited by the presence of gas carbon dioxide [125]. Analysis specified that the total viable counts ranged from $3.8-9.8\log_{10}\text{ cfu cm}^{-2}$ for aerobic cases, and $3.7-8.5\log_{10}\text{ cfu cm}^{-2}$ for MAP cases. However, for both AIR and MAP conditions, the growth rate is increased faster, as the storage temperature increases. For the case of XLD, significant changes occur only when temperature reaches at 15°C .

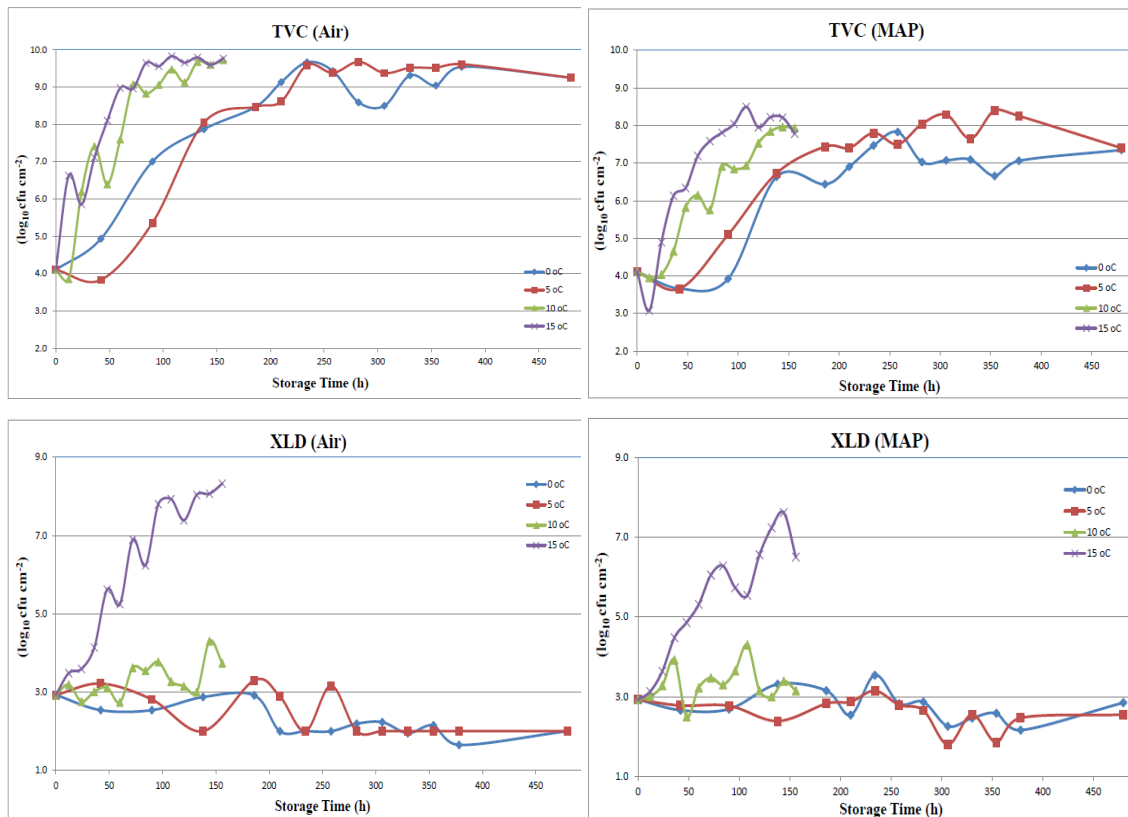


Fig. 5.4 Population dynamics of TVC and XLD at various temperatures for minced beef samples

Images from every sample (56 aerobic and 56 MAP cases) were captured using VideometerLab, (Videometer A/S, Denmark), a system which acquires multispectral images in 18 different wavelengths ranging from 405 to 970 nm. More specifically, the wavelengths are at 405, 435, 450, 470, 505, 525, 570, 590, 630, 645, 660, 700, 850, 870, 890, 910, 940 and 970 nm. Meat samples were presented in Petri dishes and collected at the same time as microbiological analysis occurred [126]. Fig. 5.5 illustrates the concept of multispectral imaging.

The acquisition system records surface reflections with a standard monochrome charge coupled device chip, nested in a calibrated digital camera. The meat sample was placed inside an Ulbricht sphere in which the camera is top-mounted. The sphere has its interior coated with matte titanium paint. The coating together with the curvature of the sphere ensures a uniform reflection of the cast light and thereby a uniform light in the entire sphere.

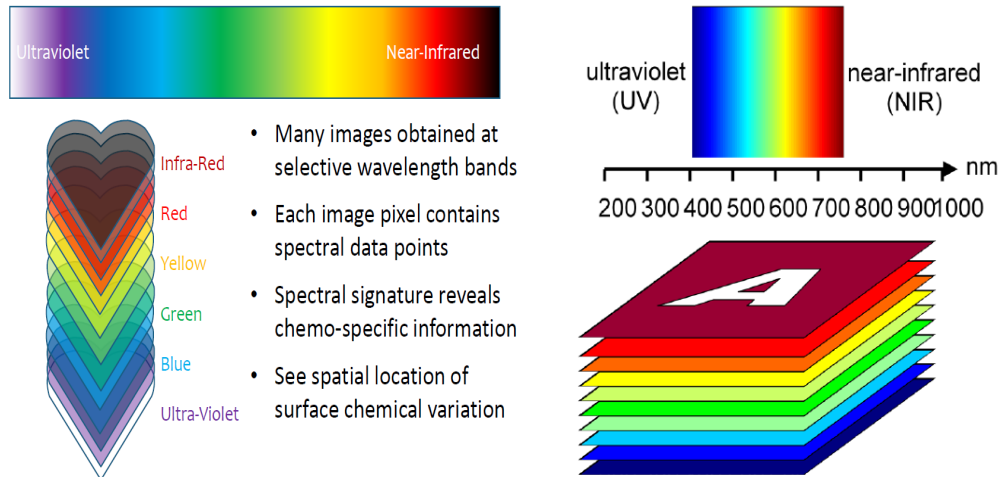


Fig. 5.5 Multispectral Imaging Concept

At the rim of the sphere, light emitting diodes (LEDs) with narrow-band spectral radiation distribution are positioned side by side in a pattern, which distributes the LEDs belonging to each wavelength uniformly around the entire rim. Each diode emits light in a specific wavelength ensuring that only light of one wavelength is present at a time. These characteristics ensure an optimal dynamic range and keep the amount of shadow and shading effects to a minimum. Each multispectral image consists of 18 separate images, one from each of the 18 wavelengths. The result is a monochrome image with 32-bit floating point precision for each LED type, giving in the end, a multispectral 3D cube of dimensionality $1280 \times 960 \times 18$ [127], as shown from Fig. 5.6.

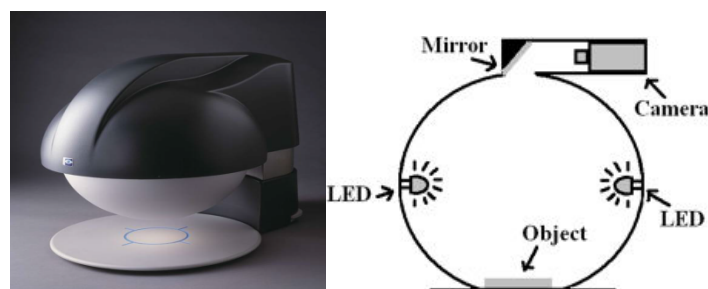


Fig. 5.6 Videometer Sensorial System

As images include redundant information, such as the Petri dish as well as meat fat, a segmentation procedure is required as a pre-processing step. The main objective of segmentation is to identify only the minced meat as the Region of Interest (ROI) from the background or any other undesired regions. This step includes transformation and segmentation procedures, which were implemented using VideometerLab software. The pre-

processing was implemented by maximizing the contrast between the sample meat material and the other non-relevant objects, enabling thus a threshold operation [128]. Canonical discriminant analysis (CDA) was employed as a supervised transformation building method to divide the images into regions of interest [129]. Following transformation using CDA, the separation was distinct and a simple thresholding was enough to separate meat from non-meat. The multispectral image sample without the background was transformed to spectrum by mean calculation. For each image, the mean reflectance spectrum was calculated by averaging the intensity of pixels within the ROI at each wavelength. Thus, the resulting data consisted of 18 mean values of the reflectance, as it was recorded by the camera for the pixels that were included in each image's ROI, and were further analysed with the proposed intelligent decision support system [123]. Fig.5.7 illustrates samples of mean reflectance spectra acquired for from both AIR and MAP minced beef samples. A close look on selected spectra at Fig. 5.7 and more precisely on the case of aerobic samples stored at 5 °C reveals that there are some differences in the reflectance's magnitude in the wavelength range from 600 nm to 850 nm, between unspoiled sample ($t = 0\text{h}$, $\text{TVC} = 4.1 \log_{10} \text{cfu cm}^{-2}$) and spoiled sample ($t = 479.5\text{h}$, $\text{TVC} = 9.3 \log_{10} \text{cfu cm}^{-2}$).

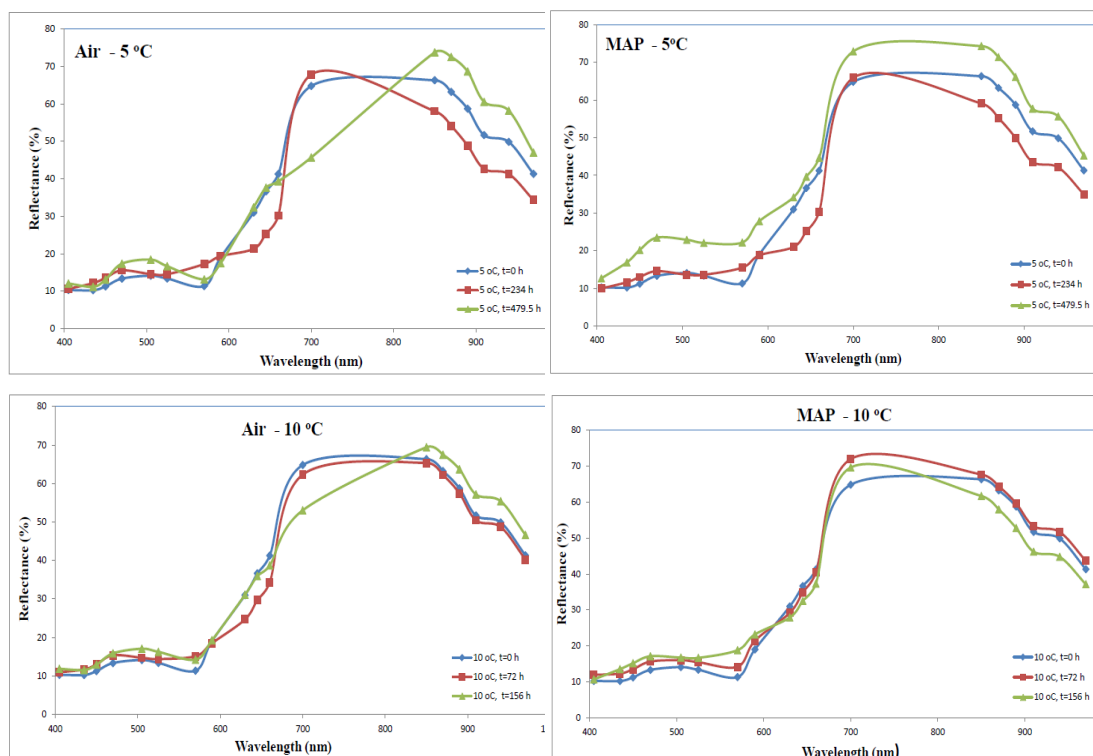


Fig. 5.7 Selected spectra for both AIR and MAP cases

These differences usually result from the spoilage and deterioration of nutrient compositions such as carbohydrates, protein, fat, which are gradually consumed and decomposed during

storage, producing a series of chemical substances, including ammonia, hydrogen sulphide, ketones and aldehydes [116]. A similar situation can be also observed in the near infrared region (850–970 nm) where reflectance values are decreased with increasing storage time [126]. Datasets related to reflectance spectra as well as the associated microbiological analysis from meat samples, were provided by Agricultural University of Athens, Greece and were further utilised towards the development of the proposed intelligent decision support system.

A principal component analysis (PCA) has been applied on those multispectral data used for training purposes. This choice was initiated mainly by two reasons. The original spectral data is considered as a high-dimensional problem, thus it could be prohibited for utilising a learning-based model (*i.e.* excessive number of input variables). Secondly, the strong correlation among the spectral variables (*i.e.* wavelengths), would deteriorate seriously the modelling procedure. In the classical PCA approach, the first component corresponds to the direction in which the projected observations have the largest variance. The second component is then orthogonal to the first component and again maximises the variance of the data points projected on it. Continuing in this way produces all of the principal components, which correspond to the eigenvectors of the empirical covariance matrix. However, both the classical variance (which is being maximised) and the classical covariance matrix (which is being decomposed) are very sensitive to anomalous observations. Consequently, the first components are often attracted toward outlying points, and may not capture the variation of the regular observations.

| PCs | Robust PCA | | |
|-----|------------|---------|--------------|
| | Eigenvalue | Prop. % | Cum. prop. % |
| 1 | 346 | 65.98 | 65.98 |
| 2 | 125 | 22.94 | 88.92 |
| 3 | 48.8 | 8.98 | 97.90 |
| 4 | 8.96 | 1.56 | 99.46 |
| 5 | 1.21 | 0.21 | 99.67 |

Table 5.1 Robust PCA scheme

An alternative approach to improve PCA appears to be the combination of projection pursuit with robust scatter matrix estimation. The goal of a robust PCA (RPCA) scheme is to obtain principal components that are not influenced much by outliers. The RPCA is implemented in three main steps. First, the data are pre-processed such that the transformed data are lying in a subspace whose dimension is at most $n-1$. A preliminary covariance matrix is then constructed and used for selecting the number of components k that will be retained in the sequel, yielding a k -dimensional subspace that fits the data well. Then the data points are projected on this subspace where their location and scatter matrix are robustly estimated, from

which its k nonzero eigenvalues l_1, \dots, l_k are computed. The corresponding eigenvectors are the k robust principal components [130]. RPCA scheme was implemented in MATLAB, with the aid of PLS_Toolbox (ver. 8.0 Eigenvector.com).

For this particular experimental case study, the first five principal components (PC) were associated with the 99.675% of the total variance, as shown in Table 5.1. These specific PCs were extracted and utilised as inputs to the various simulation models developed for this specific case study. A mandatory check however is required to validate the integrity and applicability of the developed model in predicting/classifying unknown samples to make sure that models could work in the future for new and similar data. Full cross-validation, also called leave-one-out cross-validation (LOOCV), is commonly utilized to validate the established models [131], [132]. LOOCV leaves one sample out of the calibration process, which is used for validation. All samples are used in an exhaustive way providing thus repeatability of the results compared with other random methods of partitioning of the training dataset. As the number of samples was small, separation of the dataset into training and testing subsets (hold-out method) would further reduce the number of data and would result in insufficient training of the network. Therefore, in order to improve the robustness of classification, the LOOCV method has been adopted to evaluate the performance of the developed models. Meanwhile, it is necessary to look for effective methods to evaluate the predictive effectiveness, robustness, reliability, and accuracy for practical applications. Similarly to the FTIR case, the performance of developed models for the prediction of TVC and XLD for each meat sample was determined by the bias (B_f) and accuracy (A_f) factors, the mean relative percentage residual (MRPR) and the mean absolute percentage residual (MAPR), the root mean squared error (RMSE) and finally the standard error of prediction (SEP) [132][133].

5.4 AFINN Architecture

The implemented neurofuzzy (NF) system is based on the well-known TSK type, modified however with the introduction of an additional layer of output partitions. Unlike the ANFIS NF system, where the number of local linear systems is the same as the number of rules, AFINN provides a means of controlling the growth of the number of local linear systems when the order of the system under consideration increases, so that least-squares estimation can be applied without performance degradation. A clustering algorithm is applied for the sample data in order to organize feature vectors into clusters, such that points within a cluster are closer to each other than vectors belonging to different clusters. The fuzzy rule base is derived using results obtained from a clustering algorithm.

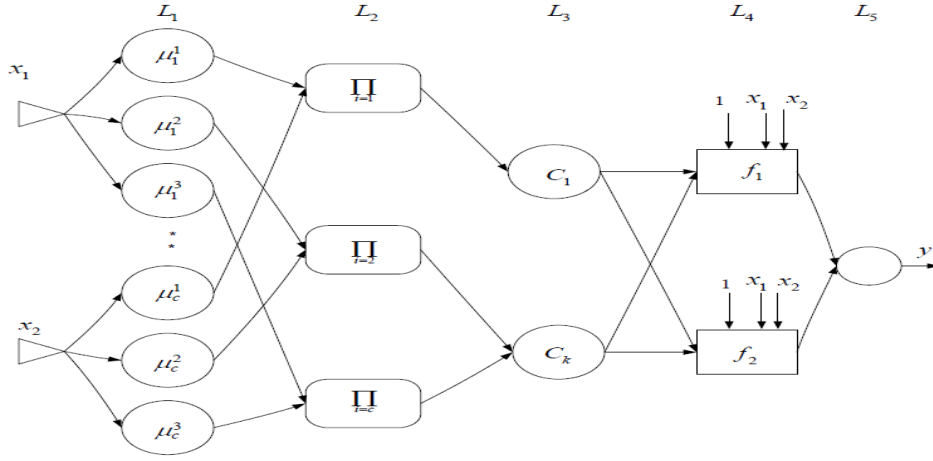


Fig. 5.8 Structure of AFINN system

The schematic of the AFINN model, shown in Fig. 5.8, consists of five layers. Layers L_1 and L_2 are associated to IF part of fuzzy rules while layers L_4 and L_5 to THEN part of these rules and are related to the defuzzification task. In layer L_3 a mapping between the rules layer and the output layer is performed through a competitive learning process and as a consequence, the linear units at L_4 are linked with each term of layer L_3 . Thus the size of required matrices for least-squares estimation at the consequent part is much smaller compared to the ANFIS approach.

5.4.1 Clustering Algorithm

The applied clustering algorithm at layer L_2 consists of two stages [134]. In the first stage the method similar to Learning Vector Quantization (LVQ) algorithm generates crisp c-partitions of the data set. The number of clusters c and the cluster centres v_i , $i = 1, \dots, c$, obtained from this stage are used by FCM (Fuzzy C-Means) algorithm in the second stage. The first stage clustering algorithm determines the number of clusters by dividing the learning data into these crisp clusters and calculates the cluster centres which are the initial values of the fuzzy cluster centres derived the second stage algorithm. Let $X = [x_1, \dots, x_n] \in \mathbb{R}^n$ be a learning data. The first cluster is created starting with the first data vector from X and the initial value of the cluster centre is taking as a value of this data vector. Then other data vectors are included into the cluster but only these ones, which satisfy the following condition

$$\|x_k - v_i\| < D \quad (5.1)$$

where $x_k \in X$, $k = 1, \dots, n$ and v_i , $i = 1, \dots, c$ are cluster centres, $V = [v_1, \dots, v_n] \in \mathbb{R}^n$, the constant value D is fixed at the beginning of the algorithm. Cluster centres v_i are modified for each cluster (i.e., $i = 1, \dots, c$) according to the following equation

$$v_i(t+1) = v_i(t) + a_i(x_k - v_i(t)) \quad (5.2)$$

where $t = 0, 1, 2, \dots$ denotes the number of iterations, $a_i \in [0, 1]$ is the learning rate and it is decreasing during performance of the algorithm (depending on the number of elements in the cluster). Recursion of Eq. 5.2, originates from the LVQ algorithm. As a result of performance of this algorithm, we get the number of clusters c , we have divided data set into the clusters, and we know values of cluster centres v_i $i = 1, \dots, c$, which we can use as initial values for the second stage clustering algorithm. In the second stage the Fuzzy C-Means algorithm has been used to optimize the values of cluster centres. Fig. 5.9 illustrates the clustering concept.

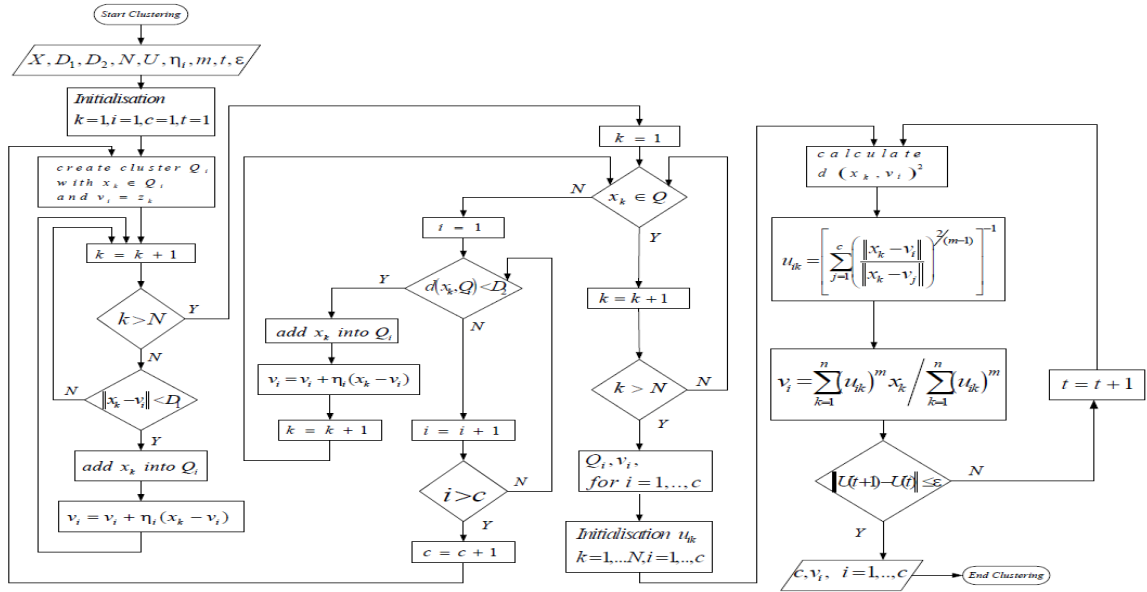


Fig. 5.9 Clustering concept

5.4.2 Feed-forward analysis of AFINN

The number of rules in the AFINN scheme is identical to the number of clusters c obtained from the clustering algorithm. Fuzzy IF-THEN rules then can be written in the following form:

$$\text{IF } (x_1 \text{ is } U_1^i \text{ AND } \dots \text{ AND } x_q \text{ is } U_q^i) \text{ THEN } (y = w_0^i + w_1^i x_1 + \dots + w_q^i x_q) \quad (5.3)$$

where U_j^i , $i = 1, \dots, c$; $j = 1, \dots, p$ and $q = p - 1$, are fuzzy sets defined based on c-partition of learning data X . The membership functions of fuzzy sets U_j^i have been chosen as Gaussian membership functions with the following form:

$$O_{U_j^i}^1 = \mu_{U_j^i} = \exp \left[- \left(\frac{x_j - v_{ij}}{\sigma_{ij}} \right)^2 \right] \quad (5.4)$$

for $j = 1, \dots, q$ and $i = 1, \dots, c$. The values v_{ij} in Eq. (5.4) represent the centres of the membership functions and are equal to the values of the components of vectors \mathbf{v}_i , which derive from the FCM algorithm. The values σ_{ij} in Eq. (5.4) define the widths of the membership functions. These values are calculated according to

$$\sigma_{ij} = \left(\sum_{k=1}^n u_{ik} (x_{kj} - v_{ij})^2 / \sum_{k=1}^n u_{ik} \right)^{1/2} \quad (5.5)$$

The second layer L_2 has c elements that realize a multiplication operation. Outputs of this layer represent the fire strength of the rules, expressed as:

$$O_i^2 = \prod_{j=1}^q O_{U_j^i}^1 \quad (5.6)$$

where $i = 1, \dots, c$. Nodes at the additional layer (L_3), represent the partitions of the output variables. The nodes should perform the fuzzy OR operation to integrate the fired rules:

$$O_l^3 = \sum_k O_k^2 w_{l,k}^3 \quad (5.7)$$

where, $k = 1, \dots, c$. Hence, links between L_2 and L_3 function as an inference engine that does not require the rule-matching process. Initially, the links at layers L_2 - L_3 are fully interconnected. However, not all the rules are necessary to the fuzzy system. The weight of the link connecting the k^{th} rule node from L_2 and the l^{th} output partition at L_3 is denoted as $w_{l,k}^3$ and assigned to be 0.5. A competitive learning algorithm is then utilised. For the set of training data pairs (x, y) the weights are adjusted as:

$$\Delta w_{l,k}^3 = O_l^3 (-w_{l,k}^3 + O_k^2) \quad (5.8)$$

where O_l^3 is denoted as the output of the l output term node, while O_k^2 is the output of the k fuzzy rule node. Hence, O_l^3 serves as a win-loss index of competition. As the competitive algorithm needs the number of output nodes O_l^3 to be a priori known, this has been heuristically set to be **(c/2+1)** of the defined number of rules c . The main principle of this phase is to remove the less important rules and to retain essential ones based on the results of competitive learning through the whole set of trained data pairs. The weight of a link that connects a rule node and an output partition node indicates the strength of the rule affecting the output partitions. The link with the maximum weight is chosen and it is assigned to 1, while the remaining ones to 0. Therefore, only the rule with the link of maximum weight will be assigned to the output partitions. After that, the weights of the links that connect the same output term node are compared. If the weight of the link is found to be small compared to the maximum one, the weight of the link is assigned to zero. The remaining weights are then

assigned to 1. Hence $w_{l,k}^3$ will be either 0 or 1, which indicates the existence of the links connecting the node l in L_3 and the node k in L_2 . At layer L_4 , every node is an adaptive node, with a node function as:

$$O_l^4 = \frac{O_l^3}{\sum_l O_l^3} f_l = \frac{O_l^3}{\sum_l O_l^3} (p_l x_1 + q_l x_2 + r_l) \quad (5.9)$$

where $\{p_l, q_l, r_l\}$ is the consequent parameter set of this node. Finally in the last layer, L_5 , the single node in this layer computes the overall output as the summation of all incoming signals:

$$O^5 = \sum_l O_l^4 \quad (5.10)$$

Similarly to the ANFIS model, a hybrid learning approach has been also adopted for the AFINN scheme [111]. All modelling schemes have been implemented in MATLAB (ver. R2014a, Mathworks.com).

5.4.3 Tuning Premise and Consequence AFINN parameters

Two different sets of AFINN parameters need to be tuned. These include the nonlinear premise parameters in the fuzzification part and the linear consequent parameters in the defuzzification part. A hybrid learning approach thus has been adopted for the AFINN scheme. In this phase, the error backpropagation is applied to tune the premise parameters of the membership functions and recursive least squares estimation is applied to find the consequence parameters of local linear systems. For each training pair (x, y) , the system output O^5 is obtained in forward pass after feeding input pattern into the network. Then the purpose of this learning phase is that, for a given p^{th} training data pair (x_p, y_p) , the parameters are adjusted so as to minimise the error function

$$E_p = \frac{1}{2} (y_p - O^5)^2 \quad (5.11)$$

The update for the premise parameters is defined as:

$$\Delta v_{jk} = -\eta \frac{\partial E_p}{\partial O_k^2} O_k^2 \left[\frac{2(x_j - v_{jk})}{(\sigma_{jk})^2} \right] \quad (5.12)$$

where η is the learning rate. The width σ is calculated as

$$\Delta \sigma_{jk} = -\eta \frac{\partial E_p}{\partial O_k^2} O_k^2 \left[\frac{2(x_j - v_{jk})^2}{(\sigma_{jk})^3} \right] \quad (5.13)$$

The recursive least-squares estimation is used to find the consequence parameters of the local linear systems.

Now let us define the estimation error e_p at the p^{th} training data pair:

$$e_p = y_p - O_p^5 = y_p - \phi_p^T \hat{\theta}_{p-1} \quad (5.14)$$

The system output is re-formulated as:

$$O^5 = \sum_l \left[\sum_m \left(\left(\frac{O_l^3}{\sum_l O_l^3} z_m \right) w_{lm} \right) \right] \quad (5.15)$$

where l is the number of nodes at L_3 , and m denote the number of input variables plus 1 (i.e., $m=1, n_1, \dots, n_q$). The consequent parameters $\{p, q, r\}$ in Eq. 5.9 are denoted as w . Let us denote,

$$\phi_p = \left[\frac{O_1^3}{\sum_l O_l^3} z_1 \dots \frac{O_1^3}{\sum_l O_l^3} z_{n_1} \frac{O_2^3}{\sum_l O_l^3} z_1 \dots \frac{O_2^3}{\sum_l O_l^3} z_{n_1} \dots \frac{O_l^3}{\sum_l O_l^3} z_1 \dots \frac{O_l^3}{\sum_l O_l^3} z_{n_q} \right] \quad (5.16)$$

and

$$\hat{\theta}_p = [w_{11} \dots w_{1n_q} w_{21} \dots w_{2n_q} w_{l1} \dots w_{ln_q}] \quad (5.17)$$

Thus, the recursive least-squares estimation can be applied to find the parameters such that the cost function J is minimised.

$$J = \frac{1}{2} \sum_p e_p^2 \quad (5.18)$$

The algorithm for updating parameters is:

$$\hat{\theta}_p = \hat{\theta}_{p-1} + \frac{P_{p-1} \phi_p}{1 + \phi_p^T P_{p-1} \phi_p} e_p \quad (5.19)$$

$$P_p = P_{p-1} - \frac{P_{p-1} \phi_p \phi_p^T P_{p-1}}{\phi_p^T P_{p-1} \phi_p} \quad (5.20)$$

where $\hat{\theta}_0$ is given and P_{-1} is an identity matrix.

5.5 Data Analysis

The final dataset consisted of 56 minced beef samples at aerobic and 56 samples at MAP conditions respectively. Information related also to sampling times was also considered for this analysis.

5.5.1 Classification of Meat Samples

The classification accuracy acquired by the AFINN model for the categorization of storage conditions (Aerobic vs. MAP) is presented in the form of a confusion matrix in Table 5.2 For this specific model, 22 rules have been created by the clustering scheme, while the input

vector consisted of the five PCs extracted from the RPCA algorithm. The hybrid parameter-learning algorithm resulted in a high speed training process, *i.e.* 20 epochs. The sensitivities reveal an overall excellent performance for both cases. The model overall achieved a 95.53% correct classification, and 96.43% and 94.64% for AIR and MAP meat samples, respectively. The sensitivities for AIR and MAP-based meat samples reveal 54 (including two marginal cases) AIR samples, and 53 MAP samples properly classified to their own class. Misclassified samples “1A5”, “1A10” correspond to minced beef AIR samples stored at 5°C and 10°C respectively and collected immediately (0h of storage). Similarly, misclassified samples “1M5”, “1M10”, “1M15” correspond to minced beef MAP samples stored at 5°C, 10°C and 15°C respectively and collected instantly (0h of storage). Such misclassification can be explained by the fact, that at $t = 0$, meat samples share the same spectral information.

| Class (AIR/MAP) | Predicted class (AFINN) | | Row total | Sensitivity (%) |
|---|-------------------------|------------|-----------|-----------------|
| | <i>AIR</i> | <i>MAP</i> | | |
| AIR (n = 56) | 52 (+2 marginal) | 2 | 56 | 96.43 |
| MAP (n = 56) | 3 | 53 | 56 | 94.64 |
| Column total (n_j) | 57 | 56 | 112 | |
| Specificity (%) | 94.74 | 94.64 | | |
| Overall correct classification (accuracy): 95.53% | | | | |

Table 5.2 Confusion matrix for class of storage conditions

The specificity index was also high, indicating satisfactory discrimination between these two classes. In addition to AFINN, an ANFIS model has been also developed to classify AIR/MAP samples. Under the same training conditions, ANFIS performed very satisfactory, its performance however was achieved with a relatively computational cost, utilising 32 fuzzy rules, using two membership functions for each input variable. An overall classification accuracy of 93.75% resulted in 7 misclassifications. In addition to previously misclassified samples, new samples “4A0” and “4M0” were also failed to be identified. These samples correspond to AIR and MAP samples stored at 0°C, collected after 138h of storage respectively.

5.5.2 Temperature Identification Model

The changes in microbial flora of fresh minced meat has been monitored at different storage temperatures (0 to 15°C) under aerobic and MAP conditions. Results from microbiological analysis, revealed that changes in Total Viable Counts follow temperature changes during storage and thus, temperature could be considered as a good indicator for meat spoilage. However, the knowledge of storage temperature is not always available, thus this issue could be considered as an obstacle for production line use.

The motivation for this research study derives from the aim to predict, for the first time, directly the storage temperature by utilising only multispectral information. Such non-invasive temperature “measurement” could be then utilised for the prediction of TVC and XLD levels.

| Temp (AIR/MAP) | Predicted class (AFINN) | | | | | | | | Row Total | Sensitivity Total (%) |
|---|-------------------------|-------|-------|-------|------|------|-------|-------|--------------|--------------------------|
| | AIR | | | | MAP | | | | | |
| | 0 °C | 5 °C | 10 °C | 15 °C | 0 °C | 5 °C | 10 °C | 15 °C | | |
| 0 °C | 13 | | 1 | | 13 | | 1 | | 28 | 92.85 |
| 5 °C | | 13 | 1 | | | 13 | 1 | | 28 | 92.85 |
| 10 °C | | 1 | 13 | | | | 14 | | 28 | 96.43 |
| 15 °C | | | 1 | 13 | | | 1 | 13 | 28 | 92.85 |
| Column total (n _j) | 13 | 14 | 16 | 13 | 13 | 13 | 17 | 13 | 112 | |
| Specificity (%) | 100 | 92.85 | 81.25 | 100 | 100 | 100 | 82.35 | 100 | | |
| Overall correct classification (accuracy): (AIR: 92.85%, MAP: 94.64%) | | | | | | | | | | 93.75% |

Table 5.3 Confusion matrix for temperature using AFINN model

The accuracy acquired by an AFINN model for the temperature prediction was 93.75% and is presented in the form of a confusion matrix in Table 5.3 Seven minced meat samples were not identified properly. These include the aerobic “1A0”, “1A5”, “5A10”, “1A15” and the MAP “1M0”, “1M5”, “1M15” samples. The “1A0”, “1A5”, “1A15” cases correspond to AIR samples stored at 0°C, 5°C and 15°C respectively and collected immediately (0h of storage). The case “5A10” corresponds to an AIR sample stored at 10°C and collected at 48h. Similarly, “1M0”, “1M5”, “1M15” cases correspond to MAP samples stored at 0°C, 5°C and 15°C respectively and collected immediately (0h of storage).

An ANFIS model has been also developed to predict temperature levels. An overall classification accuracy of 92.85% resulted in 8 misclassifications, as clearly shown in Table 5.4. In addition to the misclassified samples, which were collected immediately (0h of storage), new samples “9M5” and “13M5” were also failed to be identified. These cases correspond to MAP samples both stored at 5°C, but collected at 282h and 378h respectively. Additionally, an MLP network has been implemented using the same conditions using two hidden layers (with 24 and 12 nodes respectively). Due to the usage of gradient descent learning algorithm, 20,000 epochs were applied, resulting thus a rather slow training procedure. The prediction accuracy obtained from MLP was inferior to those achieved by both AFINN and ANFIS, with an overall rate of 90.17%.

| Temp (AIR/MAP) | Predicted class (ANFIS) | | | | | | | | Row Total | Sensitivity ANFIS (%) | Sensitivity MLP (%) |
|---|-------------------------|------|-------|-------|-------|------|-------|-------|--------------|-----------------------------|------------------------|
| | AIR | | | | MAP | | | | | | |
| | 0 °C | 5 °C | 10 °C | 15 °C | 0 °C | 5 °C | 10 °C | 15 °C | | | |
| 0 °C | 13 | | 1 | | 13 | | 1 | | 28 | 92.85 | 89.28 |
| 5 °C | | 13 | 1 | | 2 | 11 | 1 | | 28 | 85.71 | 89.28 |
| 10 °C | | | 14 | | | | 14 | | 28 | 100 | 89.28 |
| 15 °C | | | 1 | 13 | | | 1 | 13 | 28 | 92.85 | 92.85 |
| Column total (n_j) | 13 | 13 | 17 | 13 | 15 | 11 | 17 | 13 | 112 | | |
| Specificity (%) | 100 | 100 | 82.35 | 100 | 86.66 | 100 | 82.35 | 100 | | | |
| Overall correct classification (accuracy) - ANFIS: (AIR: 94.64%, MAP: 91.07%) | | | | | | | | | | 92.85% | |
| Overall correct classification (accuracy) - MLP: (AIR: 91.07%, MAP: 89.28%) | | | | | | | | | | 90.17% | |

Table 5.4 Confusion matrix for temperature using ANFIS / MLP models

5.5.3 Total Viable Counts Identification Model

AFINN models have been also constructed for TVC prediction for both Aerobic and MAP cases [135]. For each case, two simulation studies were carried out.

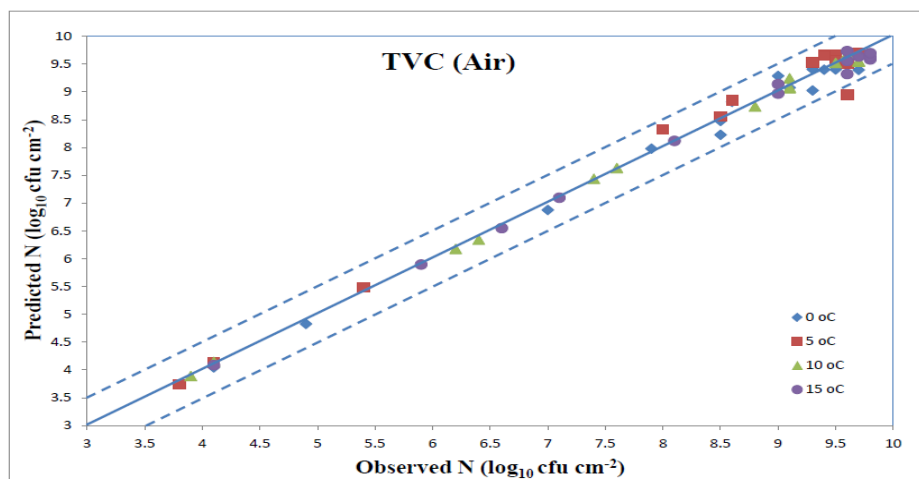


Fig. 5.10 AFINN prediction model for TVC (AIR case- all inputs)

5.5.3.1 AIR Case

In the first study, AFINN's input vector consisted of the five PCs extracted from the RPCA algorithm, as well as the sampling time and temperature information, while in the second study only the extracted PCs were considered as input variables. The number of rules used in these networks was 34 and 22 for each study respectively.

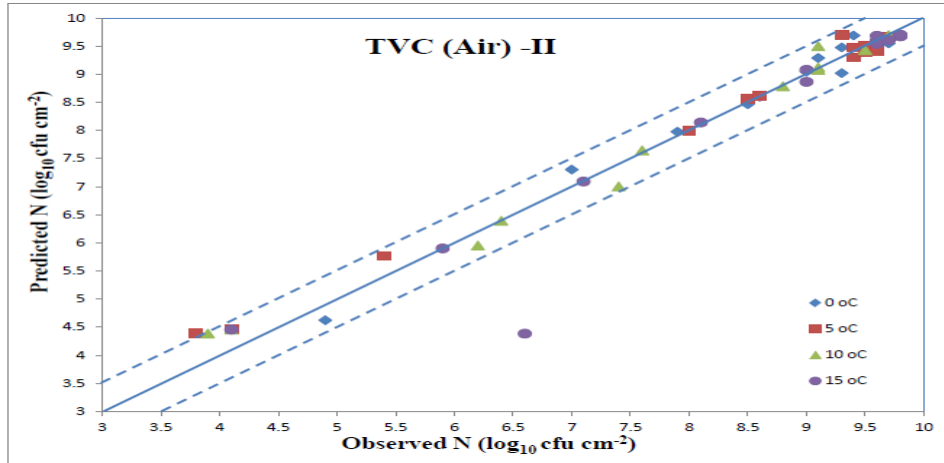


Fig. 5.11 AFINN prediction model for TVC (AIR case- RPCA inputs)

Results revealed that the identification accuracy of the AFINN model was very satisfactory in the prediction of TVCs for the AIR dataset, indicating the advantage of this approach in tackling nonlinear problems, such as meat spoilage. The plot of predicted vs. observed TVCs is illustrated in Fig. 5.10, and shows a very good distribution around the line of equity; with almost all the data included within the ± 0.5 log unit area. Based on Fig. 5.10, the “7A5” pattern that corresponds to a minced beef sample stored at 5°C and collected after 234h of storage was placed outside the specified area.

| TVC – AIR case (LOOCV) PCA inputs, time, temperature | Mathematical Expression | Temperatures (AFINN) | | | | Total AFINN | Total AFINN (46/10) |
|---|---|----------------------|--------|--------|--------|----------------|---------------------------|
| | P = the predicted values O = observed values n = number of observations | 0 °C | 5 °C | 10 °C | 15 °C | | |
| Mean squared error (MSE) | $\frac{1}{n} \sum_{i=1}^n (P - O)^2$ | 0.0304 | 0.0599 | 0.0064 | 0.0153 | 0.028 | 0.1335 |
| Root mean squared error (RMSE) | $\sqrt{\frac{\sum (O - P)^2}{n}}$ | 0.1745 | 0.2447 | 0.0797 | 0.1238 | 0.1673 | 0.3654 |
| Mean relative percentage residual (MRPR %) | $\frac{100}{n} \times \sum \frac{(O - P)}{O}$ | 0.5465 | -0.787 | 0.1028 | 0.5387 | 0.1003 | -1.5921 |
| Mean absolute percentage residual (MAPR %) | $\frac{100}{n} \times \sum \frac{ O - P }{O}$ | 1.6684 | 2.1346 | 0.7659 | 0.9869 | 1.3889 | 3.3159 |
| Bias factor (B_f) | $10^{\sum \log(P/O)/n}$ | 0.9943 | 1.0075 | 0.9989 | 0.9945 | 0.9988 | 1.0152 |
| Accuracy factor (A_f) | $10^{\sum \log(P/O)/n}$ | 1.0169 | 1.0216 | 1.0077 | 1.0100 | 1.014 | 1.0332 |
| Standard error of prediction (SEP %) | $\frac{100}{O} \sqrt{\frac{\sum (O - P)^2}{n}}$ | 2.1274 | 2.9941 | 1.0124 | 1.4728 | 2.0498 | 4.1566 |

Table 5.5 Statistical performance for AIR case (all inputs)

The performance of the AFINN model to predict TVCs in minced beef samples in terms of statistical indices is presented in Table 5.5. The RMSE values of the model were very low for testing samples, with an overall indicator of 0.1673. The accuracy factor A_f , which indicates the spread of results about the prediction, reveal that predicted total viable counts were 1.4% above from the observed values for meat samples. The mean relative percentage residual index (MRPR) verified the overall under-prediction for samples ($MRPR > 0$). Finally, the standard error of prediction (SEP) index was 2.049 % for the overall samples indicating a good performance of the network for microbial count predictions.

In order to investigate further the capabilities of AFINN model for this specific identification problem, a second experiment was carried out, where the initial multi-AIR dataset was divided into a training subset with approx. 82% of the data and a testing subset with the remaining 18% (*i.e.* 10 samples). The performance of the AFINN model to predict TVCs in minced beef samples for this second experiment, in terms of statistical indices is also presented in Table 5.5. Based on the new calculated values of the bias factor B_f , it can be assumed that model has over-estimated ($B_f > 1$) microbial population. However, a closer comparison of AFINN's performance at these two experiments reveals a problem with the limited number of samples for training. The SEP index is worse in this second case, and this reflects an open problem in learning-based systems, *i.e.* the need to have as large as possible training datasets.

| TVC – AIR case (LOOCV) PCA inputs, time, temperature | Total ANFIS | Total MLP | Total NLR | Total PLS |
|---|------------------------|----------------------|----------------------|----------------------|
| Mean squared error (MSE) | 0.0579 | 0.0744 | 0.0909 | 0.9022 |
| Root mean squared error (RMSE) | 0.2406 | 0.2727 | 0.3015 | 0.9498 |
| Mean relative percentage residual (MRPR %) | -0.2408 | -0.486 | -0.2166 | -2.396 |
| Mean absolute percentage residual (MAPR %) | 2.4768 | 3.0725 | 3.3923 | 11.263 |
| Bias factor (B_f) | 1.0019 | 1.0041 | 1.0010 | 1.0109 |
| Accuracy factor (A_f) | 1.0250 | 1.0310 | 1.0342 | 1.1105 |
| Standard error of prediction (SEP %) | 2.9479 | 3.3412 | 3.6938 | 11.6366 |

Table 5.6 Statistical performances for AIR case (all inputs- comparison)

An ANFIS and MLP models have been developed to predict TVCs utilising the same training conditions. ANFIS model performed very satisfactory, as shown in Table 5.6, its performance however was achieved with a high computational cost, utilising 128 fuzzy rules and subsequently a large number of consequent parameters. After a few trials, the MLP was constructed with two hidden layers (with 12 and 10 nodes respectively) and one output node for the TVC prediction.

The performance of the MLP network in predicting TVC in meat samples in terms of statistical indices is also presented in Table 5.6. Although both AFINN and ANFIS share the same TSK-style architecture, the clustering component allowed AFINN to achieve a superior performance. On the other hand, the localisation spread through the membership functions, is one advantage of ANFIS and AFINN models against the classic MLP structure.

In addition to these computational intelligence structures, partial least squares (PLS) and nonlinear regression schemes have been applied to the same dataset, in order to reveal the advantage of advanced learning-based methods. The PLS model was constructed using the same input vector as in the cases of AFINN, and the PLS_Toolbox software (ver. 8.0, Eigenvector.com) in association with MATLAB was used to perform the PLS analysis. The SIMPLS algorithm has been chosen as the appropriate optimisation scheme [136]. The algorithm calculates the PLS factors directly as linear combinations of the original variables. These factors are determined such as to maximize a covariance criterion, while obeying certain orthogonality and normalization restrictions. The optimal number for latent variables was set to 7. The following PLS model is associated with this specific case study.

$$Y1 = 6.63310 + 0.00555 * X1 + 0.07886 * X2 - 0.00677 * X3 + 0.05901 * X4 + 0.14028 * X5 + 0.06439 * X6 + 0.11419 * X7 \quad (5.21)$$

where X_1 represents the sampling time, X_2 the temperature, and the remaining X_i inputs the five PCs from the RPCA scheme. Nonlinear regression is often used to model complex phenomena, which cannot be handled by the linear model. The XLSTAT (v. 2015.2) software provides such capability through the use of nonlinear regression (NLR) modelling using the nonlinear iterative partial least squares (NIPALS) algorithm. For this specific case, the following 4th order model has been constructed using XLSTAT and achieved a remarkable performance compared to PLS scheme. Its performance could be easily compared to MLP's results.

$$\begin{aligned} Y1 = & 4.96194 + 0.05334 * X1 - 0.33833 * X2 - 0.00504 * X3 + 0.00893 * X4 + 0.11732 * X5 + 0.00651 * X6 \\ & - 0.11490 * X7 - 0.00026 * X1^2 + 0.07909 * X2^2 - 0.00127 * X3^2 + 0.00079 * X4^2 - 0.01233 * X5^2 \\ & - 0.02901 * X6^2 + 0.27772 * X7^2 + 6.03629E-7 * X1^3 - 0.00344 * X2^3 - 5.67724E-6 * X3^3 + 0.00001 * X4^3 \\ & - 0.00074 * X5^3 + 0.00156 * X6^3 + 0.03664 * X7^3 - 5.13162E-10 * X1^4 + 7.81010E-7 * X3^4 \\ & - 1.47147E-6 * X4^4 + 0.00006 * X5^4 + 0.00025 * X6^4 - 0.04060 * X7^4 \end{aligned} \quad (5.22)$$

Statistical information for both NLR and PLS models is illustrated at Table 5.6. However, such performance from PLS scheme was expected, as it is well known that linear PLS has some difficulties in its practical applications since most real problems are inherently nonlinear.

| TVC – AIR case (LOOCV) PCA inputs | Temperatures (AFINN) | | | | Total AFINN | Total ANFIS | Total MLP | Total NLR | Total PLS |
|--|----------------------|--------|--------|--------|----------------|----------------|--------------|--------------|--------------|
| | 0 °C | 5 °C | 10 °C | 15 °C | | | | | |
| Mean squared error (MSE) | 0.0399 | 0.0607 | 0.0535 | 0.3661 | 0.1301 | 0.1989 | 0.2564 | 0.3004 | 1.1807 |
| Root mean squared error (RMSE) | 0.1998 | 0.2463 | 0.2314 | 0.6051 | 0.3606 | 0.446 | 0.5063 | 0.5481 | 1.0866 |
| Mean relative percentage residual (MRPR %) | -0.755 | -2.208 | -1.139 | 2.0953 | -0.5018 | -0.6087 | 0.1852 | 0.7906 | -3.0959 |
| Mean absolute percentage residual (MAPR %) | 2.3601 | 3.0684 | 2.7021 | 3.6667 | 2.9493 | 4.3986 | 5.2674 | 5.5568 | 12.8970 |
| Bias factor (B_f) | 1.0070 | 1.0210 | 1.0104 | 0.9739 | 1.0029 | 1.0032 | 0.998 | 1.0046 | 1.0127 |
| Accuracy factor (A_f) | 1.0237 | 1.0299 | 1.0267 | 1.0426 | 1.0307 | 1.0455 | 1.0548 | 1.0567 | 1.1245 |
| Standard error of prediction (SEP %) | 2.4369 | 3.0141 | 2.9399 | 7.1969 | 4.4182 | 5.4645 | 6.2031 | 6.7148 | 13.3119 |

Table 5.7 Statistical performance for AIR case (PCA inputs)

For the second simulation study, the input vector was consisted of the five only PCs extracted from the RPCA algorithm. The plot of predicted vs. observed TVCs is illustrated in Fig. 5.11, and shows a good distribution around the line of equity. The comparison of Fig. 5.10 with the related Fig. 5.11 is more than evident. One sample, the “2A15”, is clearly outside the border line of the ± 0.5 log unit area and it is associated to a meat sample stored at 15°C and collected after 12h of storage. Three samples (*i.e.* “2A10”, “2A5”, “4A10”) are however in the border line of the ± 0.5 log unit area. “2A5” corresponds to a minced beef, stored at 5°C and collected after 42h of storage, while “2A10” and “4A10” were stored at 10°C and collected after 12h and 36h of storage respectively. The performance of the AFINN model to predict TVCs in minced beef samples for this second simulation, in terms of statistical indices is presented in Table 5.7. Based on the calculated values, undoubtedly the SEP index is worse in this second scenario, and this is mainly explained by the absence of the sampling time of meat samples from the input vector. There is an open problem of incorporating the time into the spectral information, which could be investigated in a future research. AFINN’s performance is still however superior to other applied models, especially against PLS which is considered as a standard modelling tool in food microbiology.

5.5.3.2 MAP Case

An important advancement in food packaging techniques is the development of Modified Atmosphere Packaging (MAP). Modified atmospheric packaged foods have become increasingly more available, as food manufactures are interested for foods with extended shelf life. In addition to aerobic TVCs prediction, AFINN models have been also applied for minced beef samples packaged under modified atmosphere conditions.

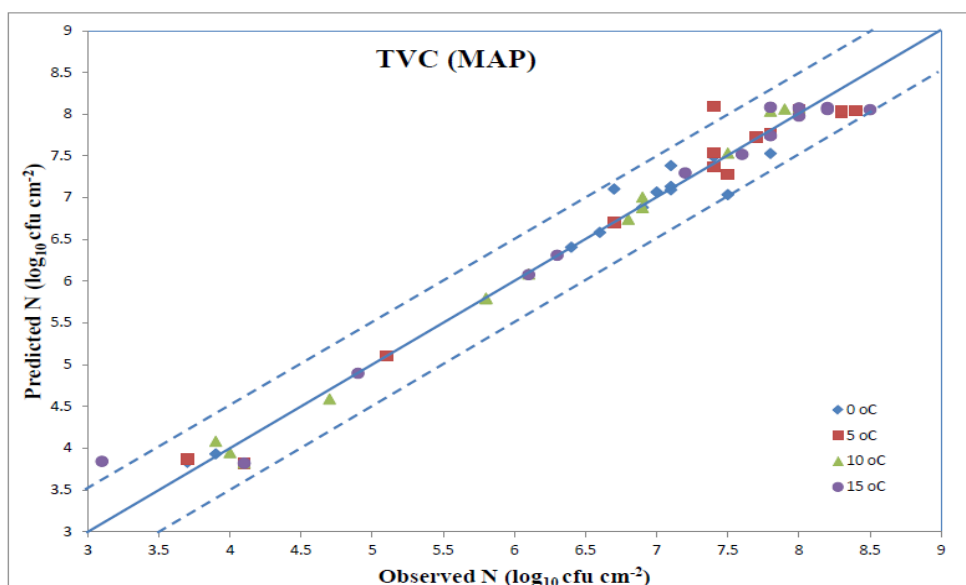


Fig. 5.12 AFINN prediction model for TVC (MAP case- all inputs)

The plot of predicted vs. observed TVCs for MAP spectra is illustrated in Fig 5.12, and shows a good distribution around the line of equity, with almost all the data included within the ± 0.5 log unit area, only for the case where additional features (i.e. sampling time, temperature) were included as input variables. Based on Fig 5.12, “2M15” and “14M5” patterns were clearly outside the borderline. “2M15” corresponds to a minced beef sample stored at 15°C and collected after 12h of storage, while “14M5” corresponds to a sample stored at 5°C and collected after 479.5h of storage. Three samples (i.e. “10M15”, “12M5”, “7M0”) were however in the border line of the ± 0.5 log unit area. “10M15” corresponds to a minced beef, stored at 15°C and collected after 108h of storage, while “12M5” was stored at 5°C and collected after 354h. Finally, meat sample “7M0” corresponds to a minced beef, stored at 0°C and collected after 234h of storage.

| Statistical index – MAP case (LOOCV) PCA inputs, time, temperature | Temperatures (AFINN) | | | | Total AFINN | Total AFINN (46/10) |
|---|----------------------|--------|--------|--------|----------------|---------------------------|
| | 0 °C | 5 °C | 10 °C | 15 °C | | |
| Mean squared error (MSE) | 0.046 | 0.0668 | 0.0163 | 0.0693 | 0.0496 | 0.0960 |
| Root mean squared error (RMSE) | 0.214 | 0.2585 | 0.1276 | 0.2632 | 0.2227 | 0.3098 |
| Mean relative percentage residual (MRPR %) | -0.021 | 0.2853 | -0.057 | -0.890 | -0.1708 | 1.7648 |
| Mean absolute percentage residual (MAPR %) | 2.446 | 2.7323 | 1.7543 | 3.4124 | 2.5863 | 3.5787 |
| Bias factor (B_f) | 0.999 | 0.9964 | 1.0002 | 1.0068 | 1.0008 | 0.9779 |
| Accuracy factor (A_f) | 1.025 | 1.0277 | 1.0178 | 1.0331 | 1.0258 | 1.0372 |
| Standard error of prediction (SEP %) | 3.362 | 3.7002 | 2.0716 | 3.8460 | 3.3784 | 4.2436 |

Table 5.8 Statistical performance for MAP case (all inputs)

The performance of the AFINN model to predict TVCs in minced beef samples for the MAP case, in terms of statistical indices is presented in Table 5.8. The RMSE values of the AFINN model were very low, with an overall indicator of 0.22. A SEP value of 3.38% was calculated for this specific study, which is however higher compared to the equivalent achieved SEP index for the AIR samples.

Overall, a comparison against AFINN's performance for AIR case, reveal an increased level of difficulty in predicting TVCs for samples packaged in MAP conditions. Similarly to the AIR case, an experiment was carried out, where the initial multi-MAP dataset was divided into a training subset with approx. 82% of the data and a testing subset with the remaining 18% (*i.e.* 10 samples). The performance of the AFINN model to predict TVCs for this experiment, in terms of statistical indices is also presented in Table 5.8. Based on the new calculated values of the bias factor B_f , it can be assumed that model has under-estimated ($B_f < 1$) microbial population, while the SEP index was increased to 4.24%. Furthermore, an ANFIS and MLP model have been developed to predict TVCs for the MAP case.

Similarly to the previous aerobic case study, both ANFIS and MLP performed very satisfactory, as shown in Table 5.9, MLP's performance however was achieved with a computational cost, by utilising two hidden layers (with 18 and 12 nodes respectively), while ANFIS model utilised 128 fuzzy rules. In addition to these learning-based structures, PLS and NLR schemes have been also applied to the same dataset.

| Statistical index – MAP case (LOOCV) PCA inputs, time, temperature | Total ANFIS | Total MLP | Total NLR | Total PLS |
|---|----------------|--------------|--------------|--------------|
| Mean squared error (MSE) | 0.0707 | 0.1103 | 0.181 | 1.0268 |
| Root mean squared error (RMSE) | 0.266 | 0.3321 | 0.4254 | 1.0133 |
| Mean relative percentage residual (MRPR %) | -0.2573 | 0.3622 | 0.6577 | -3.5658 |
| Mean absolute percentage residual (MAPR %) | 3.4666 | 4.2612 | 5.5206 | 15.3059 |
| Bias factor (B_f) | 1.0012 | 1.0015 | 1.0027 | 1.0172 |
| Accuracy factor (A_f) | 1.0349 | 1.0428 | 1.0559 | 1.1548 |
| Standard error of prediction (SEP %) | 4.0351 | 5.038 | 6.4542 | 15.3741 |

Table 5.9 Statistical performance for MAP case (all inputs- comparison)

The following PLS regression model is associated with this MAP dataset

$$Y1 = 5.01285 + 0.00516 * X1 + 0.08757 * X2 + 0.01390 * X3 + 0.07088 * X4 - 0.03170 * X5 + 0.01656 * X6 + 0.00409 * X7 \quad (5.23)$$

For this specific case, the following 5th order NLR model has been also constructed using XLSTAT 2015 and the results are also summarised at Table 5.9.

$$\begin{aligned}
Y1 = & 2.69484 + 0.02708 * X1 - 0.03163 * X2 - 0.03285 * X3 + 0.09362 * X4 - 0.02093 * X5 + 0.10341 * X6 \\
& + 0.27366 * X7 + 0.00010 * X1^2 + 0.02910 * X2^2 + 0.00084 * X3^2 - 0.00448 * X4^2 + 0.01279 * X5^2 \\
& - 0.00246 * X6^2 - 0.24896 * X7^2 - 1.14461E-6 * X1^3 - 0.00138 * X2^3 + 0.00008 * X3^3 - 0.00050 * X4^3 \\
& + 0.00020 * X5^3 - 0.00723 * X6^3 - 0.12594 * X7^3 + 2.94030E-9 * X1^4 - 4.93287E-7 * X2^4 + 0.00005 * X3^4 \\
& - 0.00009 * X4^4 - 0.00006 * X5^4 + 0.04442 * X7^4 - 3.61721E-8 * X3^5 - 1.43622E-6 * X4^5 - 2.69827E-6 * X5^5
\end{aligned}
\tag{5.24}$$

The AFINN model was also tested with the reduced input vector for this MAP study. The plot of predicted vs. observed TVCs is illustrated in Fig 5.13, and shows a distribution around the line of equity, with eleven samples placed however outside the ± 0.5 log unit area. This specific plot, compared with the equivalent for aerobic case, reveals the difficulty in predicting correctly meat samples under MAP conditions. Five patterns (*i.e.* “2M15”, “4M15”, “5M15”, “7M15”, “11M15”) were associated to meat samples stored at 15°C and collected after 12h, 36h, 48h, 72h and 120h respectively. Three patterns (*i.e.* “4M5”, “9M5”, “13M5”) were associated to meat samples stored at 5°C and collected after 138h, 282h and 378h respectively. Two patterns (*i.e.* “4M0”, “8M0”) were associated to meat samples stored at 0°C and collected after 138h and 258h respectively. Finally, one pattern, “4M10”, was associated to meat samples stored at 10°C and collected after 36h of storage.

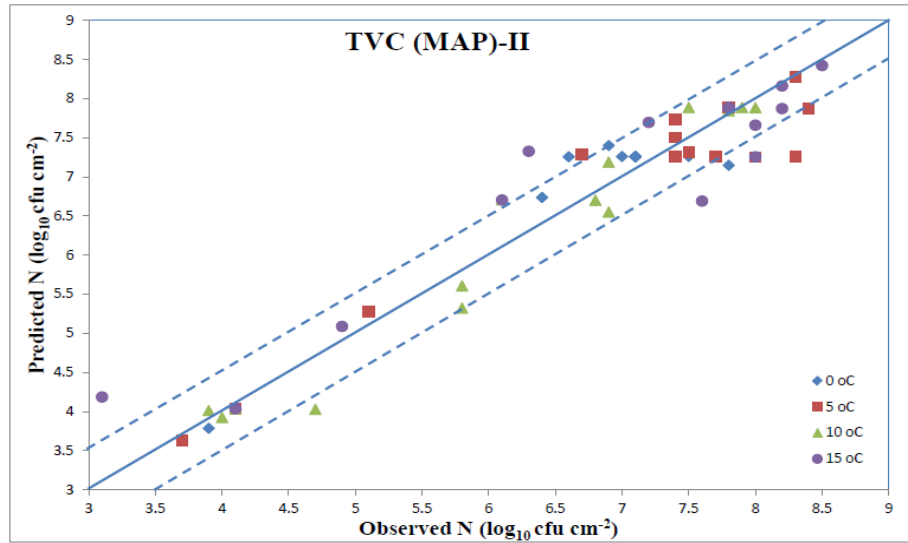


Fig 5.13 AFINN prediction model for TVC (MAP case- RPCA inputs)

The performance of AFINN model to predict TVCs in minced beef samples for this MAP case study, in terms of statistical indices is presented in Table 5.10. The sole use of PCs in the input vector resulted in a severe deterioration of the prediction accuracy, as clearly shown by all statistical indices. Table 5.10, however, reveals an additional important issue. Both neurofuzzy schemes (*i.e.* AFINN and ANFIS) managed to keep their SEP index below to 10%, while in the same time, the MLP neural network achieved a not satisfactory prediction

performance. In fact, MLP's performance could be comparable to the one achieved by the NLR scheme which has been also applied to the same dataset.

| TVC MAP case (LOOCV) - PCA inputs | Temperatures (AFINN case) | | | | Total AFINN | Total ANFIS | Total MLP | Total NLR | Total PLS |
|--|---------------------------|--------|--------|--------|----------------|----------------|--------------|--------------|--------------|
| | 0 °C | 5 °C | 10 °C | 15 °C | | | | | |
| Mean squared error (MSE) | 0.1287 | 0.1919 | 0.1062 | 0.3223 | 0.1873 | 0.3374 | 0.5844 | 0.7543 | 1.4963 |
| Root mean squared error (RMSE) | 0.3587 | 0.4380 | 0.3260 | 0.5677 | 0.4327 | 0.5808 | 0.7644 | 0.8685 | 1.2232 |
| Mean relative percentage residual (MRPR %) | 0.0436 | 1.5998 | 1.0274 | -2.998 | -0.4913 | -1.2785 | -2.8391 | -2.6597 | -4.9347 |
| Mean absolute percentage residual (MAPR %) | 4.3587 | 4.3702 | 4.2935 | 7.6113 | 5.1584 | 7.0959 | 10.5108 | 11.9987 | 18.8424 |
| Bias factor (B_f) | 1.0147 | 0.9825 | 0.9880 | 1.0243 | 1.0022 | 1.0058 | 1.0183 | 1.0131 | 1.0234 |
| Accuracy factor (A_f) | 1.0439 | 1.0456 | 1.0446 | 1.0748 | 1.0522 | 1.0692 | 1.1088 | 1.1211 | 1.1912 |
| Standard error of prediction (SEP %) | 5.6239 | 6.2704 | 5.2939 | 8.2961 | 6.5656 | 8.8126 | 11.5981 | 13.1766 | 18.5589 |

Table 5.10 Statistical performance for MAP case (PCA inputs)

5.5.4 Salmonella Identification Model

Finally, two AFINN models have been developed for the prediction of growth levels of Salmonella (XLD) for both AIR and MAP conditions. The number of rules created by the clustering unit in these two AFINN networks was 28 and 32 for AIR and MAP cases respectively. Results revealed that the accuracy of the AFINN model was very satisfactory in the prediction of XLD for the AIR dataset

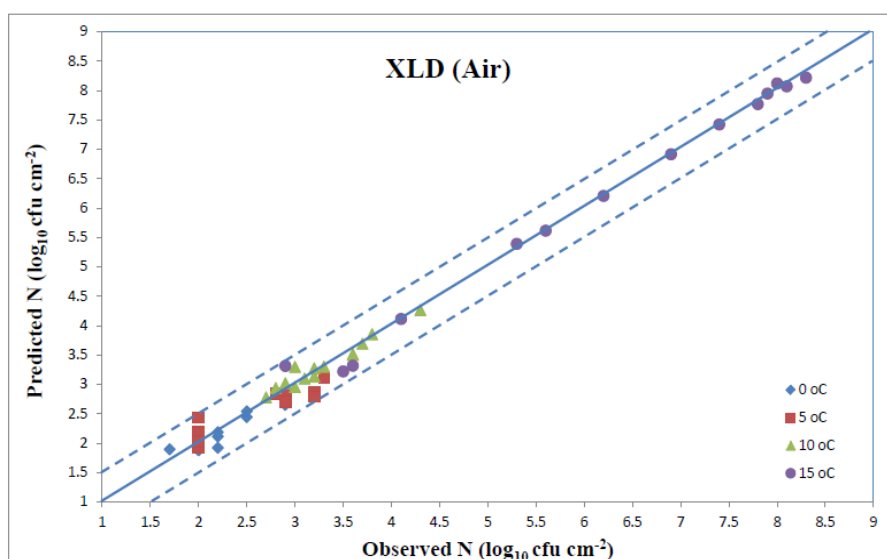


Fig. 5.14 AFINN prediction model for XLD (AIR case- all inputs)

The plot of predicted vs. observed XLD is illustrated in Fig. 5.14, and shows a very good distribution around the line of equity; with all the data included within the ± 0.5 log unit area. Based on Fig. 5.14, an excellent fitting has been achieved for the minced samples stored at 15°C and 10°C. This can be also verified through the statistical indices, which are presented in Table 5.11.

| XLD - Statistical index AIR case (LOOCV) PCA inputs, time, temperature | Temperatures (AFINN) | | | | Total AFINN | Total ANFIS | Total MLP |
|--|----------------------|--------|---------|---------|----------------|----------------|--------------|
| | 0 °C | 5 °C | 10 °C | 15 °C | | | |
| Mean squared error (MSE) | 0.0162 | 0.0476 | 0.0116 | 0.0260 | 0.025 | 0.0430 | 0.0644 |
| Root mean squared error (RMSE) | 0.1273 | 0.2183 | 0.1076 | 0.1612 | 0.159 | 0.2072 | 0.2539 |
| Mean relative percentage residual (MRPR %) | 2.4355 | -1.029 | -1.1968 | -0.1493 | 0.015 | -0.1971 | -0.7459 |
| Mean absolute percentage residual (MAPR %) | 4.4350 | 7.2118 | 2.5285 | 2.6404 | 4.204 | 5.6081 | 5.8530 |
| Bias factor (B_f) | 0.9742 | 1.0064 | 1.0114 | 1.0003 | 0.998 | 0.9992 | 1.0030 |
| Accuracy factor (A_f) | 1.0462 | 1.0736 | 1.0251 | 1.0265 | 1.043 | 1.0577 | 1.0607 |
| Standard error of prediction (SEP %) | 5.5684 | 8.9097 | 3.2613 | 2.6359 | 4.501 | 5.8586 | 7.1760 |

Table 5.11 Statistical performance for AIR case (XLD case)

Based on the calculated values, the SEP index is very low for these temperatures, while the overall SEP value is considered as acceptable for this specific problem, taking into account the XLD growth graphs at Fig. 5.14. Furthermore, ANFIS and MLP models have been developed to predict XLD for the aerobic case. Similarly to the previous aerobic case studies, both ANFIS and MLP performed very satisfactory, as shown in Table 5.11.

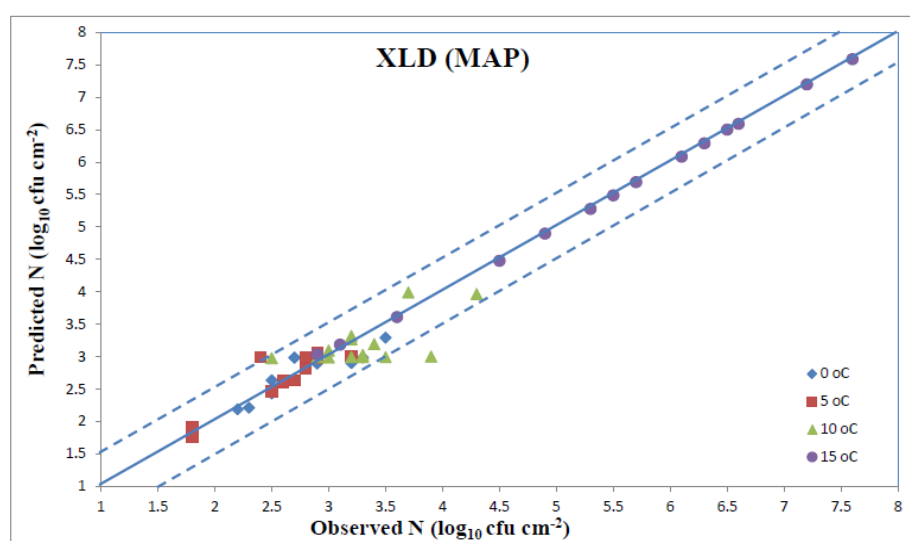


Fig. 5.15 AFINN prediction model for XLD (MAP case- all inputs)

The prediction of salmonella growth levels under MAP conditions, proved to be less accurate from the equivalent AIR case, similarly to the previous TVC predictions. The plot of predicted vs. observed XLD is illustrated in Fig. 5.15 and shows a good distribution around the line of equity, with all the data, except one, included within the ± 0.5 log unit area. Based on Fig. 5.15, an excellent fitting has been achieved for the minced samples stored at 15°C.

| XLD - Statistical index MAP case (LOOCV) PCA inputs, time, temperature | Temperatures (AFINN) | | | | Total AFINN | Total ANFIS | Total MLP |
|--|----------------------|--------|--------|--------|----------------|----------------|--------------|
| | 0 °C | 5 °C | 10 °C | 15 °C | | | |
| Mean squared error (MSE) | 0.0327 | 0.0367 | 0.1267 | 0.0020 | 0.0495 | 0.0661 | 0.1054 |
| Root mean squared error (RMSE) | 0.1808 | 0.1917 | 0.3560 | 0.0449 | 0.2226 | 0.2571 | 0.3247 |
| Mean relative percentage residual (MRPR %) | -0.477 | -3.272 | 2.5640 | -0.430 | -0.4041 | -1.076 | 0.2048 |
| Mean absolute percentage residual (MAPR %) | 4.9747 | 4.8535 | 8.2098 | 0.7047 | 4.6857 | 6.0833 | 7.9474 |
| Bias factor (B_f) | 1.0029 | 1.0305 | 0.9692 | 1.0042 | 1.0015 | 1.0071 | 0.9948 |
| Accuracy factor (A_f) | 1.0509 | 1.0473 | 1.0884 | 1.0070 | 1.048 | 1.0624 | 1.0814 |
| Standard error of prediction (SEP %) | 6.4911 | 7.3320 | 10.741 | 0.8298 | 6.3018 | 7.2788 | 9.1919 |

Table 5.12 Statistical performance for MAP case (XLD case)

The performance of the AFINN model to predict XLD in minced beef samples for this MAP case study, in terms of statistical indices is presented in Table 5.12. ANFIS model's performance was very satisfactory, achieving a comparable to AFINN's SEP prediction. Although MLP and PLS schemes have already been applied to similar multispectral / hyperspectral studies, the exploitation of neurofuzzy models for this specific imaging related application is completely novel, according to existing research/literature. Overall prediction for TVC and XLD cases has been considered as very satisfactory, although lower performance was observed especially for the MAP cases. ANFIS's prediction performance appeared to be comparable to AFINN's case; however such results were achieved with huge expensive computational cost. Prediction performances of MLP, and PLS schemes revealed the deficiencies of these systems which have been used extensively in the area of Food Microbiology.

Chapter Six

SHORT-TERM ELECTRICITY PRICE FORECASTING USING ASYMMETRIC FUZZY NEURAL NETWORKS

6.1 Introduction

Electricity, due to its nature, is difficult to store and thus, unlike other commercial products, it is not possible under normal operating conditions to keep it in stock or have customers queue for it. Additionally, power demand and supply change continuously. Hence, it is desirable for a controlling agency, such as a transmission system operator, to coordinate the dispatch of generating units to meet the expected demand of the system [137].

In the past, the centralized regulation of the electricity supply industry was considered as the best practice to guarantee security of power supply and efficient production. The power sector at that time was characterized by a highly vertically integrated market structure with very small degree of competition. However, during the last two decades, the structure of the electricity business has been transformed dramatically worldwide. Deregulated, competitive markets, where consumers have the choice to select their provider, had replaced the initial monopolistic situation. To enable trading in these new markets, exchanges and pools for electric power have been organized. Everything from real-time and spot contracts to derivatives, such as forward, future and option contracts, are traded [138].

An independent system operator (ISO) manages businesses in the electricity market. By managing various bids and offers, ISOs can obtain market trading knowledge, which is used to keep the system balance. In any electricity market, power and energy are considered commodities. Power is the rate of the transferred electric energy and is measured in megawatts (MW), while electrical energy is the energy generated by flows of electric charges and is measured in megawatt-hours (MWh). Transmission congestion and electricity derivative markets are developed in major electricity operators trading by virtue of the restructuring of electric power systems. Such restructuring, these days, is being developed in parallel with similar restructuring of natural gas markets. Modern deregulated electricity markets are generally follow day-ahead and real-time settings. In day-ahead electricity markets, demand bids and generation offers determine the electricity price of the next operating day, while in the latter case, five-minute-interval electricity prices are calculated

based on the grid operating condition. The Electricity Market Clearing price (MCP) commonly indicates the day-ahead electricity market price. Such price is determined only when the electric market is in an equilibrium state (i.e. clear of shortage and surplus). When the electricity MCP is determined, every supplier, whose offering price is below or equal to the electricity MCP, will be picked up. To maintain market fairness, and avoid corruption, all picked-up supply offers will be paid the same (MCP), regardless the offered price [139]. Short-term forecasts of the electricity price and reserve energy are issued by an ISO. Upon the public availability of this forecast, electricity suppliers and/or large-size consumers can participate (either as supplying or purchasing) via a specific bid to the ISO. The ISO then accepts initially the lowest offered price and goes up to the higher prices subjected to consumer demand's satisfaction. All approved suppliers are paid based on the last accepted offer (MCP). Hence, electricity price forecasting has become one of the most significant aspects in electricity market for trading and planning. As a result, many stakeholders are eager to invest time and money for the development of new algorithm for precise price prediction. This financial aspect has drawn great interest to research community, and has produced many significant research contributions in electricity price forecasting [140].

Electricity price forecasting models can be classified based on horizon duration, mainly into three groups, Short-Term Price Forecasting (STPF), Medium-Term Price Forecasting (MTPF) and finally Long-Term Price Forecasting (LTPF) schemes. STPF scheme is important for a quick decision making process, so that markets can design their bidding strategy using this forecasted price in order to maximize their profit in such deregulated domain. In smart grid, consumer can decide on the level of load consumption based on the current and predicted near future price. STPF approach includes next hour and day-ahead price prediction schemes [141]. MTPF scheme includes prediction for next week, next month, up to one year. The performance of this specific scheme is affected by seasonal effects, such as rise in electricity price in summer due to higher load consumption and decline in winter. MTPF information can be used by suppliers to optimize their production cost by planning an efficient resource allocation for generation of electricity. Finally, LTPF horizon varies from couple of year to decades. Such models are used by policy makers to plan pricing schemes and management of resources, while investors utilise them for analysing recovery of investment in power plant construction, production, type of energy sources and transmission [142].

Due to the importance of accurate price forecasting in volatile-style electricity market, a number of approaches have been explored in the literature. These approaches range from traditional time-series analysis to intelligent-based techniques for forecasting future prices.

A number of research studies have been performed on electricity price forecasting using the Auto Regressive Integrated Moving Average (ARIMA) method [143]. Specifically, the ARIMA method has been modified to include error correction for the worse market

conditions with high price volatility [144]. Wavelets transform and ARIMA models have been combined and used to Spanish power markets in order to improve the accuracy of price forecasting [145]. A simpler version of ARIMA Auto Regressive (AR) models have been used in Norwegian system for forecasting weekly prices [146], while ARIMA models were also utilised in Californian power market (CAISO) for forecasting daily average prices, based on historical data [147].

However, the utilisation of linear regression models in many cases, fails to capture the complicated nonlinear dynamic characteristics involved in electricity prices modelling. Thus, the use of neural networks (NN) for electricity price forecasting has come up as an improvement to above mentioned techniques. To achieve profit in the energy market it is important to predict next-day price both for consumers as well as producers for planning of electric energy resources and for developing intercession skills. Szkuta *et al.* [148] have developed NN-based models for one-step-ahead price forecasting for Victoria State in Australia, utilising historical price, load and system reserve data. Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) NNs have been utilised to forecast the average on-peak and average off-peak New England (ISONE) electricity market prices [149]. Ontario electricity market (IESO) forecasting performance was investigated through the usage of a MLP NN utilising two hidden layers [150]. Feature selection techniques and NNs have been combined to remove non-stationarity and time variance in price behaviour and this scheme has been verified on day-ahead forecasting of PJM electricity price market [151]. Recurrent NNs have been also used to eliminate complex and rough fluctuations in electricity price. An Elman recurrent NN has been used to forecast electricity prices with greater accuracy than MLP NNs. The specific methodology has been tested for robustness on the mainland Spain market data sample captured during winter week and summer weeks [152].

As an alternative to NNs, Support Vector Machines (SVM) has grown in popularity as a data-driven method. SVMs provide a non-linear mapping of the original data into high dimensional space. SVMs provide a global solution to a problem unlike MLPs which can operate within local minima of their objective function. This fact has been also recognized in many research studies related to the load and price forecasting area [153]. A two stage hybrid network of self-organized maps and SVM was explored for the ISONE electricity market [154], while improved forecasting results were also reported through the combination of Genetic algorithms with LSSVM (Least Square SVM) [155].

One of the first applications of fuzzy logic to electricity price forecasting was performed by Hong [156], who utilized fuzzy c-means for classifying historical data into three clusters (peak, medium and off-peak), and then employed a recurrent network for forecasting. A traditional fuzzy inference system utilising historical prices and previous load demand on hourly basis, has been developed to predict Marginal Cost Prices for private power vendors

in India [157]. The use of hybrid intelligent systems has been also investigated recently due to modelling advantages of such architectures. An adaptive-network-based fuzzy inference system (ANFIS) has been developed for electricity price forecasting, and results proved to be superior to equivalent MLP NN approaches [158]. A day-ahead market clearing price forecasting has been developed for the California Electricity Market through the usage of ANFIS [159]. Similarly, a wavelet neural network (WNN) has been also considered for short-term wind speed forecasting and short-term electricity price forecasting. An improved cuckoo search (CS) algorithm was proposed for optimisation of the initial weights as well as the parameters of dilation and translation in WNN model [160].

In this research, novel clustering-based neurofuzzy models are considered to compute the forecasted electricity prices in ISO New England market. In the majority of electricity price forecasting studies, especially for the hourly ahead forecasting case, only one model is usually utilized to forecast the next 24 hourly prices. However, it is a rather difficult task to associate all the characteristics of 24 different hourly prices by a single model. Thus, the model may become under-fitting for some hourly predictions, while at the same time it may become over-fitting for some others, which eventually leads to unsatisfactory results. An obvious disadvantage of such MIMO approach is related to the high complexity of the network structure (*i.e.* a system with 24 output nodes) in terms of training time and performance. Alternatively, a recurrent structure could provide similar characteristics, however in practice its performance would be deteriorated due to the feedback error accumulation. An alternative approach has been proposed in recent past [161] and it has been also adopted in this research. The core of the proposed modular forecasting system is the 24 multi-input-single-output (MISO) modelling blocks. One of the advantages of the proposed modular system is its possible use also for long-range forecasting schemes.

The proposed Asymmetric Gaussian Fuzzy Inference Neural Network (AGFINN) architecture has been designed based on the fulfilment of the following objectives

- A NF model must have minimum possible number of rules
- A NF model must be generic acting either as MISO or MIMO identification model
- A NF model must have a versatile nonlinear membership function.

Among existing neurofuzzy modelling techniques, the Takagi–Sugeno–Kang (TSK) model has attracted most attention [162]. This model consists of IF-THEN rules with fuzzy antecedents and linear functions in the consequent part. The fuzzy sets partition the input space into a number of fuzzy regions, while the consequent functions describe the system's behaviour in these regions. Clustering analysis is of great advantage as it not only defines the rules but also estimates the initial membership function parameters for the inputs simultaneously. Both TSK-based structure as well as the “clustering of fuzzy rules” concept

has been considered in the proposed AGFINN architecture. The Fuzzy C-Means (FCM) clustering has been applied as a pre-processing scheme to keep minimum number of fuzzy rules. Although a FCM algorithm has been currently integrated in the AGFINN framework, any other unsupervised clustering scheme can be also incorporated as a future work. It is well known that ANFIS scheme has issues with excessive number of rules. In the proposed scheme, the number of fuzzy memberships for each input variable is directly associated to the number of rules, hence, the “*curse of dimensionality*” problem is significantly reduced. In spite of the extensive use of the standard symmetric Gaussian membership functions, AGFINN utilizes an asymmetric function acting as input linguistic node. Since the asymmetric Gaussian membership function’s variability and flexibility are higher than the traditional one, it can partition input space more effectively [163]. AGFINN has been designed either as an MISO or MIMO system. In the MISO case, a TSK defuzzification scheme has been implemented. Two different learning algorithms have been adopted

- Gradient Descent (GD) algorithm for both premise and consequent parameters
- RLS algorithm for consequent and GD for premise parameters respectively

For the MIMO case, a “centre average” (CA) defuzzifier has been used as defuzzification scheme while the GD algorithm was utilised as learning approach.

In the following discussion and results sections, STPF results corresponding to hours with the maximum (22h) and minimum (4h) electricity prices are considered. The proposed modelling schemes are compared against AFLS, ANFIS, Wavelet network (WNN) and MLP NN modelling schemes in order to assess their prediction performance. Such comparison is considered as an essential practice, as we have to emphasise the need of induction to the area of power forecasting, advanced learning-based modelling schemes, which may have a significant potential for accurate assessment of electricity price prediction. Such an accurate assessment/prediction could allow an ISO operator to have a more efficient management of power supply.

6.2 The STPF Case Problem

ISO New England (ISONE) (<http://www.iso-ne.com>) is a not-for-profit independent corporation created in 1997 to carry out three primary tasks: to manage the daily operation of the regional power grid, to develop and oversee a market for wholesale electricity generation, and finally to ensure a reliable source of electricity to the New England region through system and market efficient planning. ISONE is the system operator for the six-state region of New England (i.e. Connecticut, Rhode Island, Massachusetts, Vermont, New Hampshire, and most of Maine). A centralized approach to managing power flow, allows for scheduling

required plant maintenance without concern of insufficient peak period generation. In order to improve system reliability and to mitigate price volatility, ISONE manages Day-ahead and Forward capacity markets as a means to schedule efficiently future supply and demand requirements. On the supply side, ISONE can call upon generating units that range from small “peak-load” units, to medium-sized units used in the presence of a quick increase in demand, and to large base-load units that are online nearly continuously. In total, ISONE has control on over 350 electricity generators within the region as well as ties among neighbouring regional grids in New York, Quebec, and New Brunswick. New England has approximately 30,500 MWs of installed electricity generating capacity and the power generation resource mix is transitioning from coal, oil, and nuclear power to natural gas and renewable energy. In total, ISONE serves 7.1 million retail electricity customers in a population of 14.7 million. Power demand is not constant, with high electricity peak of 28,000 MWs in the summer period, to smaller peak of 22,800MW in the winter.

| Hour | Max | Min | Mean | Median | STD | Coefficient of Variation |
|------|--------------|--------------|--------------------|---------------|--------------------|--------------------------|
| 1 | 82.93 | 33.82 | 51.39694521 | 50.69 | 8.508092838 | 0.165536936 |
| 2 | 79.89 | 32 | 53.5769589 | 55.135 | 9.811868548 | 0.183135974 |
| 3 | 75.7 | 29.85 | 50.38982192 | 51.69 | 9.376865197 | 0.186086492 |
| 4 | 64.99 | 20.22 | 45.42739726 | 46.7 | 9.134124929 | 0.20107084 |
| 5 | 106.55 | 35.79 | 59.43954795 | 57.78 | 13.66979665 | 0.229978139 |
| 6 | 92.41 | 37.88 | 59.72078082 | 58.3 | 9.678234207 | 0.162058065 |
| 7 | 78.37 | 29.73 | 57.64484932 | 60.825 | 10.88807959 | 0.18888209 |
| 8 | 161.79 | 35.64 | 87.09231507 | 82.175 | 24.11532545 | 0.276893839 |
| 9 | 166.93 | 49.7 | 112.7451918 | 117.02 | 20.79239258 | 0.184419329 |
| 10 | 169.31 | 38.22 | 86.92812329 | 83.995 | 21.99826678 | 0.253062714 |
| 11 | 96.75 | 0 | 59.91643836 | 59.615 | 10.81548758 | 0.180509521 |
| 12 | 105.24 | 24.36 | 53.78390411 | 55.265 | 13.18375994 | 0.245124636 |
| 13 | 217.43 | 29 | 65.62735616 | 63.915 | 23.73682855 | 0.361691068 |
| 14 | 109.06 | 28.4 | 64.8880411 | 67.17 | 14.59885589 | 0.224985308 |
| 15 | 146.46 | 25.18 | 63.23605479 | 59.98 | 18.76144335 | 0.296689023 |
| 16 | 111.89 | 32.56 | 70.46687671 | 71.56 | 14.10779615 | 0.200204647 |
| 17 | 118.1 | 26.62 | 63.02356164 | 63.61 | 15.35079884 | 0.243572379 |
| 18 | 102.51 | 29.36 | 59.14246575 | 59.355 | 14.36492454 | 0.242886805 |
| 19 | 114.11 | 30.1 | 63.35526027 | 63.86 | 12.62266848 | 0.199236313 |
| 20 | 166.78 | 35.55 | 82.46321918 | 78.675 | 24.39192868 | 0.295791614 |
| 21 | 131.91 | 57.08 | 90.1630274 | 91.02 | 15.76953729 | 0.174900264 |
| 22 | 220.1 | 60.83 | 113.0527397 | 109.96 | 27.39522212 | 0.242322496 |
| 23 | 107.18 | 40.46 | 65.62654795 | 66.545 | 10.78044055 | 0.164269505 |
| 24 | 98.73 | 35.23 | 60.60347945 | 59.655 | 10.67550225 | 0.176153289 |

Table 6.1 Statistical indices of the ISONE dataset (2006-2007)

In this work, the training/testing dataset was created from the hourly basis EP in the period 2006-2007, starting from 1/1/2016 for 723 days. Both training and testing sets were organised into 24 time series, each one corresponding to a different hour of the day. More specifically,

600 data were allocated to training subset, while 123 data for the testing one. The data used for case studies in chapter 6 are provided from a public database, found in the following link:

<https://www.iso-ne.com/isoexpress/web/reports/pricing/-/tree/zone-info>

Table 6.1 illustrates some statistical characteristics of the hourly prices of the dataset. According to mean and median prices, it is noticeable that prices are low in early morning hours and from 05:00 there is a continuous increment until 09:00. After a short decline at 10:00, a continuous increment is observed until 22:00. The coefficient of variation is an indication about the dispersion of values around the means. Low values correspond to low variability. The values vary between 0.16 and 0.36, with the majority of values around 0.20. This means that hourly values are characterized by a medium variability.

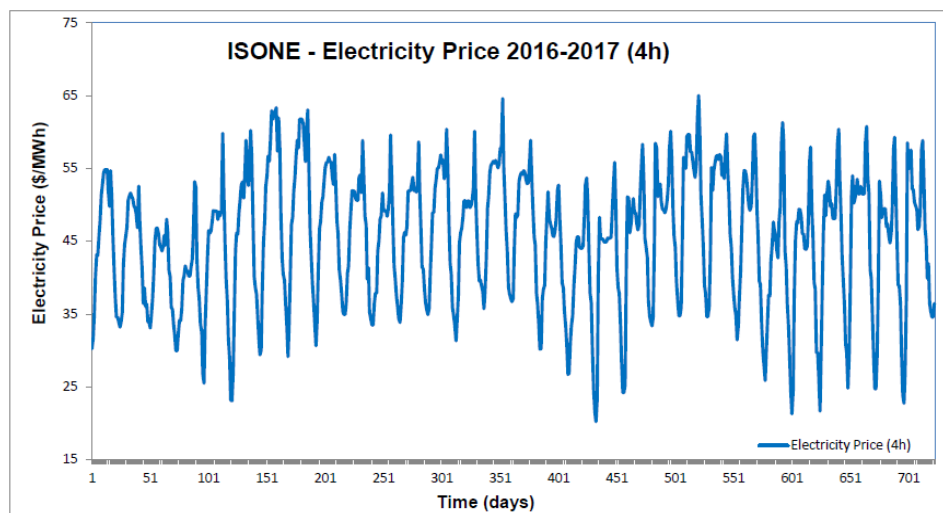


Fig. 6.1 ISONE Electricity price time series corresponding to 4h

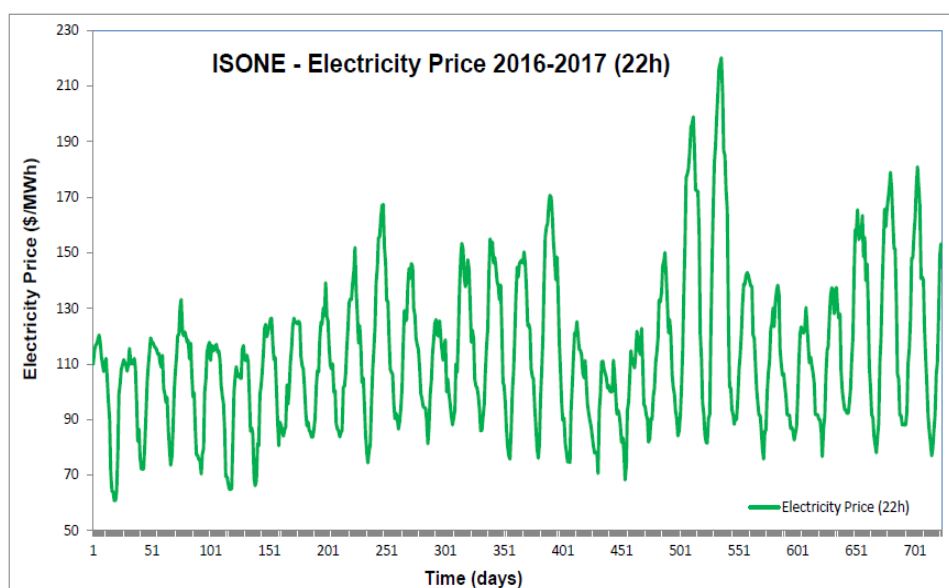


Fig. 6.2 ISONE Electricity price time series corresponding to 22h

In the following result section, only values that correspond to hours with the maximum (22h) and minimum (4h) electricity prices are considered. The ISONE time-series for minimum and maximum electricity price are shown in Figs. 6.1 and 6.2 respectively.

Electricity price forecasting is a nonlinear problem with many input variables, including past own values as well as past and forecasted values of any exogenous variables such as electricity consumption. To deal with this fact, three different models have been considered for this study, in order to extract conclusions about the most appropriate forecasting scheme in terms of input selection. In general, historical values of the parameter under study have been considered as input candidates for forecasting problems. In electricity price analysis, load factor has been considered as the most important external variable. Therefore, in this research, we assume that next day's forecasted load is also available. There is a similarity between price and load parameters. While the load level rises, a constant increase of price is observed too.

6.3 Asymmetric Neurofuzzy Model (AGFINN)

In this section, the proposed Asymmetric Gaussian Fuzzy Inference Neural Network (AGFINN) concept is presented as an alternative neurofuzzy modelling approach. Initially, AGFINN has been implemented as a MIMO neurofuzzy (NF) network, which incorporates a clustering pre-processing stage. The architecture of the proposed scheme, shown in Fig 6.3, includes also a FCM clustering scheme for structural / initialization purposes.

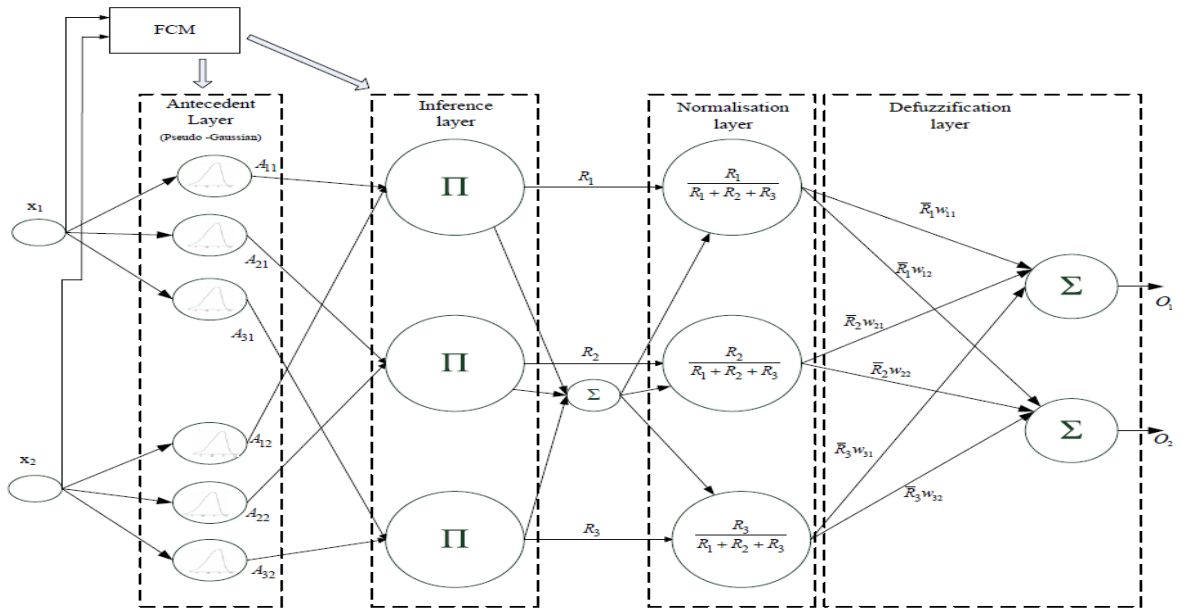


Fig. 6.3 Structure of AGFINN-CA System

In this MIMO configuration, AGFINN has been optimized through the gradient descent learning algorithm, while “centre average” (CA) defuzzifier has been used as defuzzification method. This technique is more efficient in terms of implementation compared to the traditional, for fuzzy logic systems, “centroid of area” approach [164].

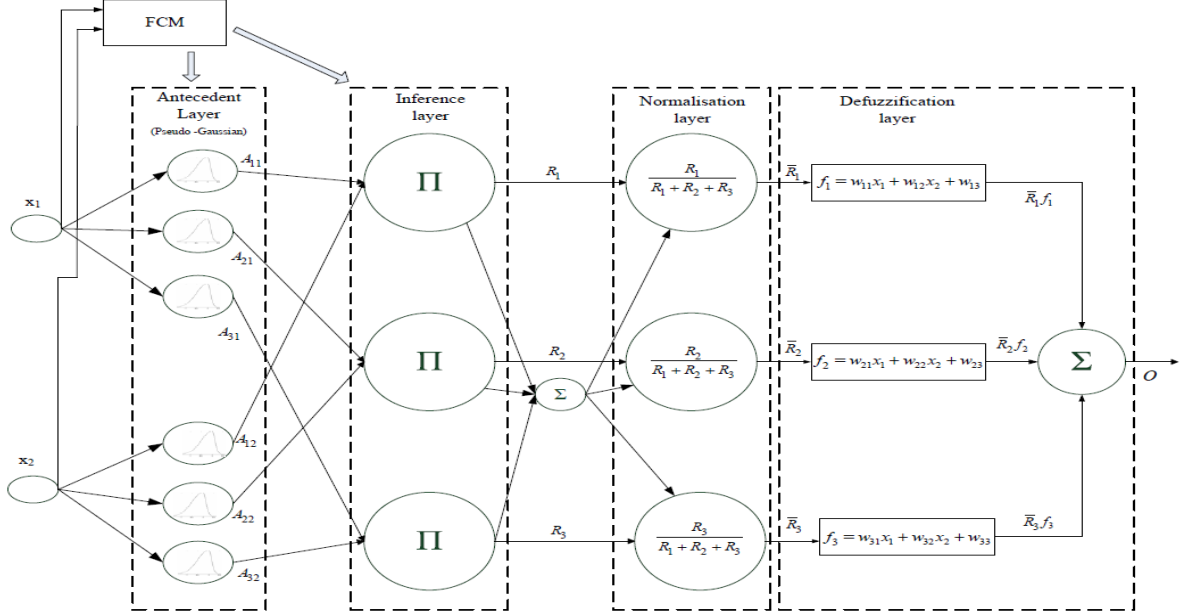


Fig. 6.4 Structure of AGFINN-TSK System

Many neurofuzzy schemes are following the TSK defuzzification style, where only one output is enabled (i.e. MISO configuration). ANFIS is a well-known representative of TSK-based neurofuzzy systems. Generally, TSK-based models allow us to model nonlinear behaviour with relatively fast training speed. Thus, it would be also interesting to investigate a TSK-based version for AGFINN and explore any possible improvement against ANFIS. Similarly to previous AGFINN-CA scheme, the AGFINN-TSK has been built around five layers, utilising the same learning algorithm. The architecture for AGFINN-TSK is shown in Fig 6.4. The first three layers L_1 , L_2 and L_3 correspond to IF part of fuzzy rules whereas layer L_5 contains information about THEN part of these rules and perform the defuzzification task. In layer L_4 a normalization process is performed for all rules derived from L_3 .

6.3.1 FCM Clustering Algorithm

Fuzzy C-Means (FCM) clustering is the most prominent fuzzy unsupervised clustering algorithm which is based on minimizing an objective function that represents the distance from any given data point to a cluster centre weighted by that data point’s membership value.

Given n data patterns, x_1, x_2, \dots, x_n , fuzzy clustering means partitioning the data patterns into c clusters which centred at c_i . The objective function for FCM is defined by

$$\sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^m d_{ij}^2, \quad 1 \leq i \leq c \quad (6.1)$$

where μ_{ij} is the degree of membership of object j in cluster i , m is the weighting exponent varying in the range $[1, \infty]$ and d_{ij} denotes the Euclidean distance between x_j and c_i . The membership μ_{ij} and the cluster centres \vec{c}_i^r are calculated by the following equations:

$$\mu_{ij} = \left[\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}} \right]^{-1}, \quad 1 \leq i \leq c, \quad 1 \leq j \leq n \quad (6.2)$$

$$\vec{c}_i^r = \frac{\sum_{j=1}^n \mu_{ij}^m \vec{x}_j}{\sum_{j=1}^n \mu_{ij}^m} \quad (6.3)$$

where $\vec{c}_i^r = [c_{i1}, \dots, c_{iq}]$ represents the multidimensional c_i cluster vector and q equals to the number of input variables.

FCM clustering is an iterative procedure, which updates c_i using the last iteration's membership values. This algorithm moves objects between clusters until the objective function cannot be decreased further. The result is a set of clusters that are as compact and well separated as possible. In the present study, cluster centres have been utilized as initial values for the centres of Asymmetric Gaussian membership functions, while the number of If-THEN rules for AGFINN modelling is equal to the number of clusters obtained through FCM clustering approach. The spread values for each membership function σ_{ij} are initialized according to

$$\sigma_{ij} = \left(\sum_{k=1}^n \mu_{ik} (x_{kj} - c_{ij})^2 / \sum_{k=1}^n \mu_{ik} \right)^{1/2} \quad (6.4)$$

These values are calculated based on the matrix U , where its elements correspond to the fuzzy memberships of input x_k in the i^{th} cluster and have centre values obtained again from FCM.

6.3.2 Feed-forward Analysis of AGFINN

The clustering algorithm provides the fuzzy c-partition of the sample data. This result helps us to generate the fuzzy rules base for AGFINN schemes. Fuzzy IF-THEN rules can be written in the following general form, based on the configuration:

$$\text{IF } (x_1 \text{ is } U_1^i \text{ AND } \dots \text{ AND } x_q \text{ is } U_q^i) \text{ THEN } \begin{cases} y_k = \sum_{j=1}^c w_{kj} R_j, & (\text{MIMO-CA}) \\ y = \sum_{j=1}^c f_j R_j, \quad f_j = w_{j1}x_1 + \dots + w_{jn}x_n + w_{j(n+1)} & (\text{MISO-TSK}) \end{cases} \quad (6.5)$$

where U are fuzzy sets defined based on c -partition of learning data X and R_c are the fuzzy normalised rules.

The structure of AGFINN schemes is explained below layer by layer:

Layer 1: This layer is simply the input layer. Nodes in this layer pass on the input signals x_1, x_2, \dots, x_n to L_2 .

Layer 2: This layer is the fuzzification layer, and its nodes represent the fuzzy sets used in the antecedent parts of the fuzzy rules. A fuzzification node receives an input and determines the degree to which this input belongs to in the node's fuzzy set. This Layer utilises an asymmetric Gaussian membership function (MF), which has the following general form

$$A_{ij} = \exp \left(-\frac{1}{2} \left(\frac{x_i - c_{ij}}{\sigma_{ij}^{\text{left}}} \right)^2 \right) U(x_i; -\infty, c_{ij}) + \exp \left(-\frac{1}{2} \left(\frac{x_i - c_{ij}}{\sigma_{ij}^{\text{right}}} \right)^2 \right) U(x_i; c_{ij}, \infty) \quad (6.6)$$

where $U(x_i; a, b) = \begin{cases} 1 & \text{if } a \leq x_i < b \\ 0 & \text{otherwise} \end{cases}$

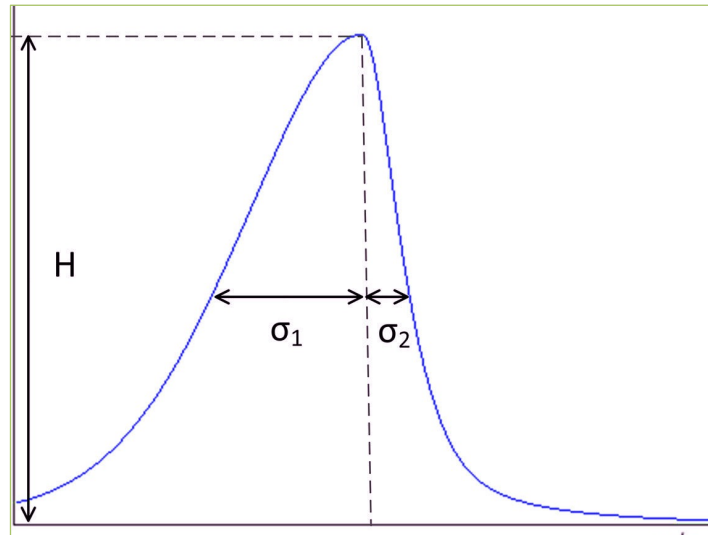


Fig. 6.5 Structure of Asymmetric MF

From the above equation, it is obvious that the proposed MF utilizes two spreads, namely $\sigma_{ij}^{\text{left}}$ and $\sigma_{ij}^{\text{right}}$ respectively. Both of these parameters transform the traditional “standard” Gaussian

function to a more asymmetric style, which can provide greater flexibility from the original one. A schematic of the proposed asymmetric MF is shown in Fig. 6.5.

Due to the asymmetric nature of the MF, both spread initialisation and updates need to be modified from the “traditional way” gradient descent algorithm that is used to update standard Gaussian MFs. Following the FCM clustering stage, an initial value for spreads ($\sigma_{ij}^{\text{init}}$) have been provided. However, AGFINN utilise two spreads, one located at the left of the initial centre parameter and one at the right. Initially, both spreads are initialised as $\sigma_{ij}^{\text{init}} / 2 = \sigma_{ij}^{\text{left}} = \sigma_{ij}^{\text{right}}$. Thus, during the first iteration of the training process, the spread of any asymmetric MF will be equal to $\sigma_{ij}^{\text{total}} = \sigma_{ij}^{\text{left}} + \sigma_{ij}^{\text{right}}$. Upon the arrival of any input variable from L_1 , there is a need to identify its position (either left or right) against the specific centre parameter for each MF. This is recorded via a specific MF index allocated for each MF. This index is then used in the backward phase to update that particular spread parameter, and is updated accordingly to any new input arrival from L_1 . During forward training phase, $\sigma_{ij}^{\text{total}}$ is used as the spread used in the Gaussian function which has the specific form

$$A_{ij} = \exp \left(-\frac{1}{2} \left(\frac{x_i - c_{ij}}{b_{ij}} \right)^2 \right) \quad (6.7)$$

where $2b_{ij} = \sigma_{ij}^{\text{total}}$, i represent the number of MF/rules, while j denotes the specific input variable. During the backward training phase, a new spread σ_{ij}^{new} value is obtained via the GD learning method. Based on the information stored at that specific MF index, either the left or right is updated as $\sigma_{ij}^{\text{left or right}} = \sigma_{ij}^{\text{new}} / 2$. For the next iteration step, in the forward training phase, the spread parameter will be equal again as $\sigma_{ij}^{\text{total}} = \sigma_{ij}^{\text{left}} + \sigma_{ij}^{\text{right}}$, incorporating however the relative adjustment of one of its components.

Layer 3: This layer is the firing strength calculation layer. Since each fuzzy rule’s antecedent part has AND connection operator, the firing strengths are calculated using the product T-norm operator. The most commonly used fuzzy AND operations are intersection and algebraic product [165]. In this case, the multiplication has been used, and the output of this layer has the following form:

$$R_i = \prod_j^n A_{ji}(x_j) \quad (6.8)$$

The number of nodes, at this layer, is equal to the number of clusters, as it was defined by the clustering pre-processing step.

Layer 4: This layer is the normalization layer. Each node in this layer calculates the normalized activation firing of each rule by:

$$\bar{R}_i = \frac{R_i}{\sum_{j=1}^c R_j} \quad (6.9)$$

The normalized activation firing is the ratio of the activation firing of a given combination to the sum of activation firings of all combinations. It represents the contribution of a given combination to the final result.

Layer 5: This layer is related to the defuzzification /output part of the AGFINN. Each node at this layer combines the output of each node in L_4 by algebraic sum operation after being multiplied by the output weight value of either f_j or w_{ij} , based on the network's configuration. Thus, the output has the following form

$$\begin{cases} O_i = \sum_{j=1}^c w_{ij} \bar{R}_j, & (MIMO-CA) \\ O = \sum_{j=1}^c f_j \bar{R}_j, & (MISO-TSK) \end{cases} \quad (6.10)$$

where $f_j = w_{j1}x_1 + \dots + w_{jn}x_n + w_{j(n+1)}$ represent the “consequent parameters” of the TSK-style defuzzification scheme.

6.3.3 AGFINN Learning Phase

The learning algorithms of AGFINN schemes involve the use of the gradient descent (GD) method to optimize the various network parameters for both AGFINN-CA and AGFINN-TSK configurations. In addition, for the AGFINN-TSK scheme, a hybrid learning algorithm, which includes the use of recursive least squares method for the consequent parameters, has been also considered in this research. During, the backward “training” passes, the error signals are calculated from the output layer backward to the premise (i.e. membership) layers, and parameters at both defuzzification and fuzzification sections are fine-tuned. For each training pair (x, y) , the system output O_i is obtained by forward pass after feeding an input pattern into the network. Then the purpose of this learning phase is that, for a given p^{th} training data pair (x_p, y_p) , the parameters are adjusted so as to minimise the error function

$$E = \frac{1}{2} \sum_{k=1}^K (D_k - O_k)^2 \quad (6.11)$$

where K is the number of outputs and D_k the desired response of the k^{th} output. Variable O_k is defined as in Eq. 6.10. According to the GD method, the weights in the defuzzification layer for the MIMO-CA scheme are updated by the following equation

$$\Delta W_{ki} = -\frac{\partial E}{\partial W_{ki}} = -\frac{\partial E}{\partial O_k} \frac{\partial O_k}{\partial W_{ki}} = (D_k - O_k) \bar{R}_i \quad (6.12)$$

where $k=1$ and $i=1,2,..c$ denote the number of outputs and normalisation units, respectively. The weights of the output units are updated according to the following equation

$$W_{ki}(t+1) = W_{ki}(t) + \eta_w \Delta W_{ki} \quad (6.13)$$

where η_w is the learning rate. For the AGFINN-TSK configuration, GD has been modified to address the parameters $f_i = w_{i1}x_1 + \dots + w_{in}x_n + w_{i(n+1)}$ in the consequent linear part of the network. Thus, the chain rule is extended to

$$\left\{ \begin{array}{l} \frac{\partial E}{\partial w_{i1}} = \frac{\partial E}{\partial O} \frac{\partial O}{\partial f_i} \frac{\partial f_i}{\partial w_{i1}} \\ \dots \\ \frac{\partial E}{\partial w_{in}} = \frac{\partial E}{\partial O} \frac{\partial O}{\partial f_i} \frac{\partial f_i}{\partial w_{in}} \\ \frac{\partial E}{\partial w_{i(n+1)}} = \frac{\partial E}{\partial O} \frac{\partial O}{\partial f_i} \frac{\partial f_i}{\partial w_{i(n+1)}} \end{array} \right. \quad \text{where } \frac{\partial O}{\partial f_i} = \bar{R}_i \quad \text{and} \quad \left\{ \begin{array}{l} \frac{\partial f_i}{\partial w_{i1}} = x_1 \\ \dots \\ \frac{\partial f_i}{\partial w_{in}} = x_{1n} \\ \frac{\partial f_i}{\partial w_{i(n+1)}} = 1 \end{array} \right. \quad (6.14)$$

The c_{ij} and b_{ij} parameters of the asymmetric membership function are adjusted by the amount

$$\begin{aligned} c_{ij}(t+1) &= c_{ij}(t) - \eta_c \left(\frac{\partial E}{\partial c_{ij}} \right) \\ b_{ij}(t+1) &= b_{ij}(t) - \eta_b \left(\frac{\partial E}{\partial b_{ij}} \right) \end{aligned} \quad (6.15)$$

$\frac{\partial E}{\partial c_{ij}}$, $\frac{\partial E}{\partial b_{ij}}$ components need to be calculated using the chain rule.

$$\begin{aligned} \frac{\partial E}{\partial c_{ij}} &= \frac{\partial E}{\partial O} \frac{\partial O}{\partial R_i} \frac{\partial R_i}{\partial A_{ij}} \frac{\partial A_{ij}}{\partial c_{ij}} \\ \frac{\partial E}{\partial b_{ij}} &= \frac{\partial E}{\partial O} \frac{\partial O}{\partial R_i} \frac{\partial R_i}{\partial A_{ij}} \frac{\partial A_{ij}}{\partial b_{ij}} \end{aligned} \quad (6.16)$$

Analytically, the partial derivatives are defined as

$$\begin{aligned} \frac{\partial A_{ij}}{\partial c_{ij}} &= \frac{\partial}{\partial c_{ij}} \left[\exp \left(-\frac{(x - c_{ij})^2}{2b_{ij}^2} \right) \right] = \left[\exp \left(-\frac{(x - c_{ij})^2}{2b_{ij}^2} \right) \right] \cdot \frac{\partial}{\partial c_{ij}} \left(-\frac{(x - c_{ij})^2}{2b_{ij}^2} \right) \\ &= A_{ij} \cdot \left(-\frac{1}{2b_{ij}^2} \left(\frac{\partial}{\partial c_{ij}} (x - c_{ij})^2 \right) \right) = A_{ij} \cdot \left(\frac{1}{b_{ij}^2} (x - c_{ij}) \right) \end{aligned} \quad (6.17)$$

$$\begin{aligned}
\frac{\partial A_{ij}}{\partial b_{ij}} &= \frac{\partial}{\partial b_{ij}} \left[\exp \left(-\frac{(x-c_{ij})^2}{2b_{ij}^2} \right) \right] = \left[\exp \left(-\frac{(x-c_{ij})^2}{2b_{ij}^2} \right) \right] \cdot \frac{\partial}{\partial b_{ij}} \left(-\frac{(x-c_{ij})^2}{2b_{ij}^2} \right) \\
&= A_{ij} \cdot \left(-\frac{(x-c_{ij})^2}{2} \cdot \frac{\partial}{\partial b_{ij}} \left(\frac{1}{b_{ij}^2} \right) \right) = A_{ij} \cdot \left(\frac{(x-c_{ij})^2}{b_{ij}^3} \right)
\end{aligned} \tag{6.18}$$

$$\frac{\partial R_i}{\partial A_{ij}} = \prod_{lm \neq ij} A_{lm} \tag{6.19}$$

$$\begin{aligned}
\frac{\partial O}{\partial R_i} &= \frac{w_i \left(\sum_{\substack{j=1 \\ j \neq i}}^c R_j \right) - \sum_{\substack{j=1 \\ j \neq i}}^c R_j w_j}{\left(\sum_{j=1}^c R_j \right)^2} \quad (\text{MIMO-CA}) \\
\frac{\partial O}{\partial R_i} &= \frac{f_i \left(\sum_{\substack{j=1 \\ j \neq i}}^c R_j \right) - \sum_{\substack{j=1 \\ j \neq i}}^c R_j f_j}{\left(\sum_{j=1}^c R_j \right)^2} \quad (\text{MISO-TSK})
\end{aligned} \tag{6.20}$$

6.3.3.1 AGFINN-TSK Hybrid Learning Scheme

Since the training of NF systems usually involves intensive computation, they are often trained off-line using techniques such as the GD method. However, the convergence of the estimated parameters using this method is often slow, as it is derived mainly from the consideration of the convergence of the training algorithm, rather than the learning speed. TSK-based NF systems utilise a linear combination of weights and input variables in their defuzzification part, hence well-established linear parameter estimation algorithms can be used to estimate these “defuzzification” weights of the model [162]. Among them, least-squares (LS) based algorithms provide a simple adaptive scheme which is capable of a fast convergence rate, good estimation accuracy and fast tracking ability to system parameter changes. In nonlinear models, that utilise LS methods, the system’s output can be written in the form of a regression model as $Y = \Phi^T \theta$, where Φ is the called regressor and θ represents the parameter vector.

The basic idea in LS estimation is to find parameters that minimise the square of the difference between the estimated and true nonlinear functions. While the batch version of LS method has proven to be very successful for a variety of applications, it is by its very nature a “batch” approach (*i.e.*, all the data are gathered, then processing is done). For small number of patterns, we could easily apply the batch calculation for increasingly more data as they are gathered, but the computations can become prohibitive due to the computation of the inverse of $\Phi^T \Phi$. Recursive version of the batch least squares method (RLS) will allow us to update our θ estimate parameter vectors each time we get a new data pair, without using all the old data in the computation and without having to compute the inverse of $\Phi^T \Phi$ [166]. The system output of the AGFINN-TSK scheme, shown in Eq. 6.10, is re-formulated as:

$$O = \sum_{l=1}^c \sum_{j=1}^{(n+1)} \left((\bar{R}_l Z_j) w_{lj} \right) \quad (6.21)$$

where l is the number of nodes at L_4 , and Z denote the number of input variables plus 1 (*i.e.* $Z = [x_1, x_2, \dots, x_n, 1]$). The consequent linear parameters are denoted as w_{ij} . Let us denote,

$$\Phi = [\bar{R}_1 x_1, \bar{R}_1 x_2, \dots, \bar{R}_1 x_n, \bar{R}_1, \dots, \bar{R}_c x_1, \bar{R}_c x_2, \dots, \bar{R}_c x_n, \bar{R}_c] \quad (6.22)$$

and

$$\theta = [w_{11}, w_{12}, \dots, w_{1n}, w_{1(n+1)}, \dots, w_{c1}, w_{c2}, \dots, w_{cn}, w_{c(n+1)}] \quad (6.23)$$

the vector parameters to be utilised in the RLS estimation method. With a new input sample $\Phi(t)$, and desired output value $d(t)$, RLS procedure is summarised as:

- Update the input history vector $\Phi(t)$
- Compute the model output using the previous set of model linear coefficients $\theta(t-1)$

$$y(t) = \Phi^T(t) \theta(t-1) \quad (6.24)$$

- Compute the error

$$e(t) = d(t) - y(t) \quad (6.25)$$

- Compute the gain vector

$$K(t) = P(t) \Phi(t) \quad (6.26)$$

where

$$P(t) = \frac{1}{\lambda} \left[P(t-1) - \frac{P(t-1) \Phi(t) \Phi^T(t) P(t-1)}{\lambda + \Phi^T(t) P(t-1) \Phi(t)} \right] \quad (6.27)$$

- Update the linear coefficients for the next iteration

$$\theta(t) = \theta(t-1) + K(t) e(t) \quad (6.28)$$

The value of weighting factor λ is defined as the system memory, and affects the convergence and the ability of the method to track time-varying statistics in the input sequence. In this research, this parameter has been set constant to one. Initialisation of $P(0)$ has been set as $P(0) = aI$ for some large $a > 0$. All modelling schemes have been implemented in MATLAB (ver. R2016a, Mathworks.com).

6.4 Results and Discussion

In order to evaluate the proposed AGFINN NF models, utilizing either CA or TSK defuzzification schemes, various simulations were carried out based on the same ISONE datasets. Input variable selection is an extremely important issue for such investigation. Historical information of electricity prices and past load demand constitutes important inputs for predicting the electricity price. The output of electricity price can take several durations, namely hourly, daily and weekly forecasting. In our investigation, the proposed forecasting architecture has been based on an hourly-horizon plan. In all our models, only historical prices and forecasted demands have been chosen as appropriate input variables, while three case studies based on different input selection schemes have been considered for all used learning-based methods. A series of trials has been required to define the optimal configuration for the structure of each learning-based model, such as number of clusters, number of neurons at hidden layer, epochs, etc. To assess the forecasting performance of the all models, a number of error measures have been employed as assessment criteria. These include root mean square error (RMSE), mean absolute percentage error (MAPE), standard error of prediction (SEP), accuracy factor (A_f), mean absolute error (MAE), Theil U1 statistic (U_1), absolute percentage error (APE) and the coefficient of determination (R^2) [167].

The RMSE index depends on the scale of the dependent variable. It should be used as relative measure to compare forecasts for the same series across different models. The smaller the RMSE index is, the better the forecasting ability of that model. However, one potential problem associated with RMSE index is the fact that the forecast error variance varies across time, due to model's nonlinearity. RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{m=1}^M (O_m - P_m)^2}{M}}$$

The Absolute Percentage Error (APE) provides information about the errors dispersion around zero. It is defined as

$$APE = 100 \times \sum_{m=1}^M \frac{|O_m - P_m|}{O_m}$$

The MAPE index provides information about the average deviation from the observed value. However, MAPE has been criticized for the problem of asymmetry and instability in cases when the original value is small. In general, MAPE index as an accuracy measure is affected with the presence of outliers, which may distort the comparisons in real case studies. MAPE is defined as

$$MAPE = \frac{100}{M} \times \sum_{m=1}^M \frac{|O_m - P_m|}{|O_m|}$$

The MAE index is also relied on the scale of the dependent variable but it is less sensitive to large deviations than the usual squared loss. MAE is defined as

$$MAE = \frac{1}{M} \sum_{m=1}^M |O_m - P_m|$$

The SEP index is determined as the relative deviation of the mean prediction values and it has the advantage of being independent on the magnitude of the measurements [164]. SEP is defined as

$$SEP = \frac{100}{\bar{O}} \sqrt{\frac{\sum_{m=1}^M (O_m - P_m)^2}{M}}$$

The idea of introducing relative measures is to evaluate the performance of a forecast relative to that of a benchmark forecast. Measures may produce very big numbers due to outliers and/or inappropriate modelling, which in turn make the comparison of different forecasts not feasible or not reliable. Relative measures may eliminate the bias introduced by potential trends, seasonal components and outliers, provided that the benchmark forecast handles these issues appropriately. One of these relative measures is the Theil U_1 coefficient. This Theil coefficient is scale invariant and it lies between zero and one. If U_1 equals zero then we have a perfect fit. This index is defined as

$$U_1 = \frac{\sqrt{\frac{1}{M} \sum_{m=1}^M (O_m - P_m)^2}}{\sqrt{\frac{1}{M} \sum_{m=1}^M (O_m)^2} + \sqrt{\frac{1}{M} \sum_{m=1}^M (P_m)^2}}$$

R-squared (R^2) is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. It measures the proportion of the variance in the dependent variable that is predictable from the independent variables. The higher the value ($0 \leq R^2 \leq 1$), the better is the prediction by the model [164].

This index is defined as

$$R^2 = 1 - \frac{\sum_{m=1}^M (O_m - P_m)^2}{\sum_{m=1}^M (O_m - \bar{O})^2}$$

Results from AGFINN schemes are compared against models based on AFLS, ANFIS, Wavelet NNs (WNN) and Multilayer Neural Networks (MLP). Such comparison is considered as a essential test, as we have to emphasise the need of induction to the area of energy systems, advanced learning-based modelling schemes, which may have a significant potential for the accurate estimation of either load or price forecasting.

6.4.1 Case Study 1

The objective of this model is to examine a simple configuration, used by various researchers, where electricity prices at previous days and hours, as well as forecasted (for the targeted hour/day) load demand are utilized as input variables. Thus, for electricity price modelling for a specific hour (i) and day (j), the following five input variables have been considered:

Target:

- *Price(i,j): electricity price at the i^{th} hour on the $(j)^{th}$ day,*

Inputs:

- *Price(i, j-1): price at the i^{th} hour on the $(j-1)^{th}$ day,*
- *Price(i, j-2): price at the i^{th} hour on the $(j-2)^{th}$ day,*
- *Price(i-1, j-1): price at the $(i-1)^{th}$ hour on the $(j-1)^{th}$ day,*
- *Price(i-2, j-1): price at the $(i-2)^{th}$ hour on the $(j-1)^{th}$ day,*
- *Load(i,j): electricity load at the i^{th} hour on the j^{th} day,*

This proposed input structure has been constructed based on the knowledge of the price at the (i,j-1) position as the main variable, while additional variables were selected from its direct “neighbourhood”. More specifically, prices on the same hour (i) at two previous days and prices on the same day (j-1) at two previous hours have been considered. In this case study, it is also assumed that the forecasted demand (i,j) is also available via a separate load forecasting model. Based on this configuration, AGFINN models have been involved in forecasting the maximum (22h) and minimum (4h) price respectively.

AGFINN-RLS models produced best results. Their optimal structure included 20 fuzzy rules for the case of 22h, while 15 rules were adequate for the case of 4h. The combination of GD +RLS, as learning algorithms, resulted in a fast training process, as the training time was completed in less than 500 epochs.

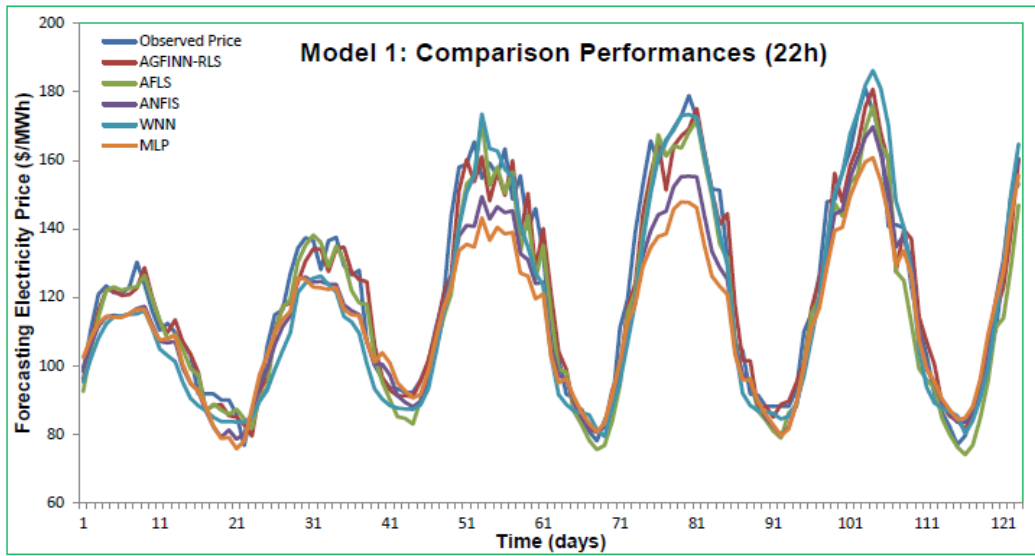


Fig. 6.6 Forecasting Electricity Price at 22h (Case Study 1)

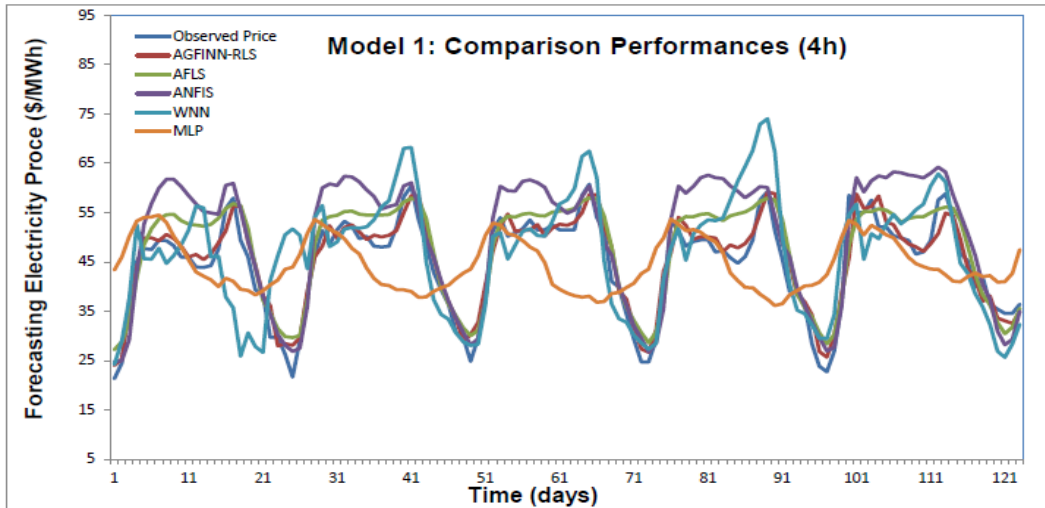


Fig. 6.7 Forecasting Electricity Price at 4h (Case Study 1)

Figs. 6.6 and 6.7 illustrate the testing performances for minimum (4h) and maximum (22h) electricity price forecasting using case study 1, for all simulated forecasting models, while Tables 6.2 and 6.3 summarise their statistical performances. Viewing these two curves, we can conclude that AGFINN-RLS model fitting with historical price data managed to forecast the result very satisfactory. Although both curves have different dynamic characteristics, it is important to mention the existence of volatile price spikes mainly for 22h case.

From the statistical tables, we can see the clear advantages of AGFINN-based models against other models implemented for this specific problem. Case Study 1 will be used as a benchmark for all implemented models for the next two case studies, where alternative input variable schemes would be presented. It is also important to mention the performance differences between AGFINN models. This is illustrated at Fig.6.8, where TSK-based

defuzzification structures surpasses CA configuration. In fact, AGFINN-CA's performance could be considered to be close to AFLS's one. Based on these indices, AGFINN schemes achieved a very good performance, especially for the case of maximum price (i.e. 22h). In order to evaluate the goodness of the current performance of the proposed AGFINN schemes, a comparison against NN, WNN and neurofuzzy models that have been employed for the specific datasets has been carried out. More specifically, AGFINN schemes have been compared against a multilayer perceptron (MLP), wavelet NN and neurofuzzy (NF) ANFIS and AFLS systems.

| | AGFINN RLS | AGFINN TSK | AGFINN CA | AFLS | ANFIS | WNN | MLP |
|----------------------|-----------------------|-----------------------|----------------------|-------------|--------------|------------|------------|
| | | | | | | | |
| RMSE | 3.1753 | 3.3783 | 4.0542 | 4.6194 | 7.5882 | 8.5394 | 10.4905 |
| APE | 727.0175 | 862.1091 | 956.1771 | 1169.1 | 1675.5 | 1892.9 | 2627.2 |
| MAPE | 5.9107 | 7.0090 | 7.7738 | 9.5050 | 13.6217 | 15.3897 | 21.3592 |
| MAE | 2.4390 | 2.7909 | 3.3526 | 3.9456 | 6.1493 | 6.2892 | 8.1517 |
| SEP | 7.1039 | 7.5578 | 9.0700 | 10.3345 | 16.9763 | 19.1044 | 23.4693 |
| U₁ | 0.0344 | 0.0369 | 0.0433 | 0.0490 | 0.0779 | 0.0908 | 0.1158 |
| R² | 0.9516 | 0.9395 | 0.9482 | 0.9325 | 0.9147 | 0.7252 | 0.1321 |

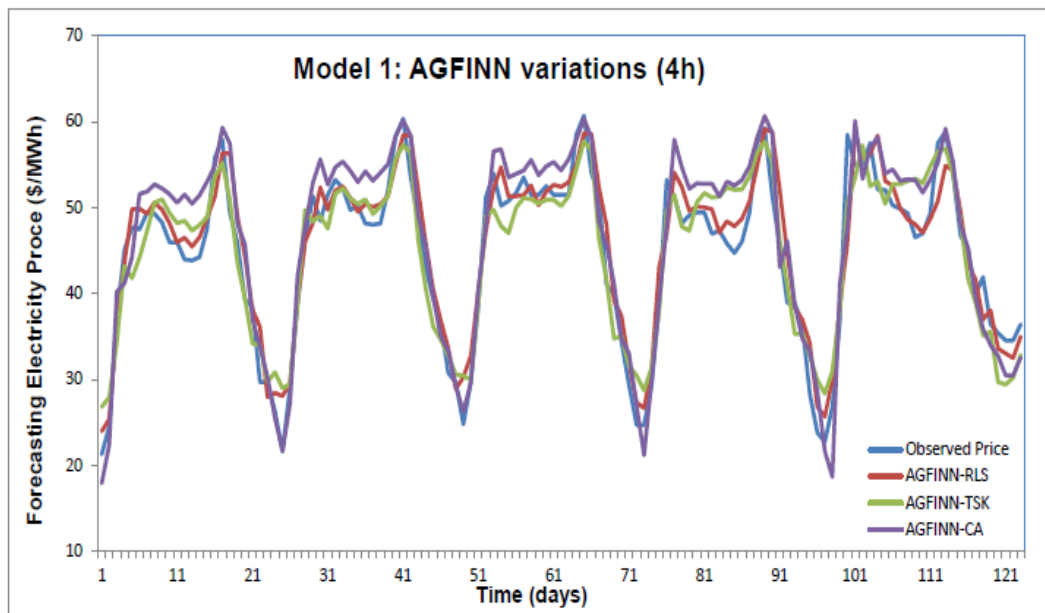
Table 6.2 Statistical performance for Electricity Price Forecasting Models at 4h

| | AGFINN RLS | AGFINN TSK | AGFINN CA | AFLS | ANFIS | WNN | MLP |
|----------------------|-----------------------|-----------------------|----------------------|-------------|--------------|------------|------------|
| | | | | | | | |
| RMSE | 7.2511 | 7.9028 | 8.1100 | 8.6060 | 10.3569 | 9.2798 | 12.8880 |
| APE | 551.0082 | 619.7377 | 607.4207 | 671.4143 | 775.0515 | 755.1376 | 912.2255 |
| MAPE | 4.4797 | 5.0385 | 4.9384 | 5.4587 | 6.3012 | 6.1393 | 7.4165 |
| MAE | 5.5696 | 6.1820 | 6.3175 | 6.7172 | 8.1925 | 7.4155 | 9.8865 |
| SEP | 5.9924 | 6.5310 | 6.7022 | 7.1122 | 8.5591 | 7.6690 | 10.6508 |
| U₁ | 0.0293 | 0.0322 | 0.0332 | 0.0353 | 0.0430 | 0.0380 | 0.0539 |
| R² | 0.9679 | 0.9671 | 0.9729 | 0.9664 | 0.9694 | 0.9639 | 0.9619 |

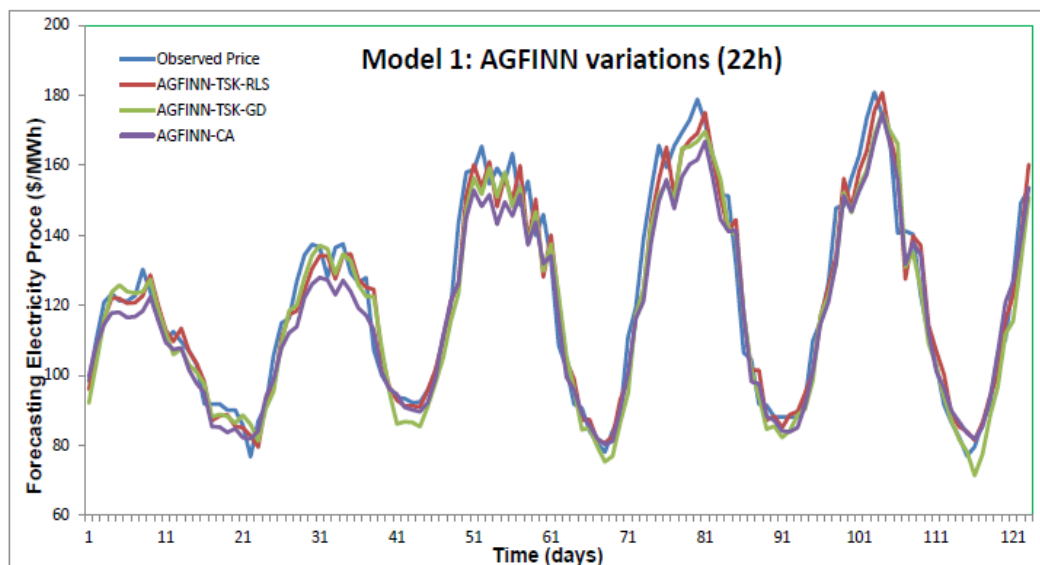
Table 6.3 Statistical performance for Electricity Price Forecasting Models at 22h

The “Adaptive Fuzzy Logic System” (AFLS) model is an advanced MIMO NF systems which includes the CA defuzzification scheme, while differs from conventional fuzzy rule-table approaches that utilize the “look-up table” concept [164]. The AFLS scheme does not follow TSK's architecture, as the number of memberships for each input variable is directly associated to the number of rules, hence, the “curse of dimensionality” problem is significantly reduced. The fuzzification component in AFLS is similar to AGFINN, with the exception of the FCM clustering step as well as the absence of asymmetric MFs. For this

specific case study, 25 fuzzy rules for the case of 22h, and 20 rules for the case of 4h were used as a final configuration.



(a)



(b)

Fig. 6.8 Comparison of AGFINN-based models (Case Study 1)

An MLP network was also constructed for this case study, using the same input vector. After a few trials, utilizing different internal structures, a NN was implemented with two hidden layers (with 20 and 8 nodes respectively). Although AGFINN-TSK, AGFINN-CA, AFLS and MLP share the same learning training algorithm, the different “philosophy” in building the neurofuzzy architecture, allowed those systems to achieve a superior performance.

An ANFIS NF model has been constructed, utilising 32 fuzzy rules. As the number of MFs in AGFINNs is equal to the numbers of rules, the proposed architecture has advantages over the classic ANFIS model. The increased number of Gaussian membership functions increases the localization of the input signal while in the same time maintains the required number of rules at low level. ANFIS's training utilises a hybrid learning algorithm, the same as the AGFINN-RLS scheme. An interesting outcome from this case study is related to the performance of wavelet neural network (WNN), which outperformed ANFIS for the case of 22h. The idea of utilising wavelets in neural networks has been proposed by Zhang & Benveniste [168]. In this type of network, all dilation and translation parameters as well as output layer weights are adjustable via GD learning algorithm. Obviously, to improve the approximation accuracy, large numbers of wavelet neurons are required for WNN with fixed wavelet bases. This may result in a large complex network structure and cause possible over-fitting problems. For this research, the wavelet function adopted in WNN's hidden layer nodes is a modified differentiable version of Morlet wavelet, which has the following form

$$\varphi_{m_j, n_j}(x_i) = \cos(2\pi\beta(\frac{x_i - m_j}{n_j}))e^{-\frac{(\frac{x_i - m_j}{n_j})^2}{\theta}}$$

where m and n are associated with the dilation and translation parameters respectively. After trials, 20 modified Morlet wavelet functions were utilised in the construction of WNN [169].

6.4.2 Case Study 2

Research has indicated that current hour electricity price shows a high correlation with those of hour h-24 and h-168, a fact that indicates daily and weekly periodicity. The objective of this particular case study is to investigate this issue, therefore no exogenous input variables have been considered as potential input variables. Thus, for electricity price modelling for a specific hour (i) and day (j), the following six input variables have been considered:

Target:

- $Price(i, j)$: electricity price at the i^{th} hour on the $(j)^{th}$ day,

Inputs:

- $Price(i, j-1)$: price at the i^{th} hour on the $(j-1)^{th}$ day,
- $Price(i, j-2)$: price at the i^{th} hour on the $(j-2)^{th}$ day,
- $Price(i, j-3)$: price at the i^{th} hour on the $(j-3)^{th}$ day,
- $Price(i, j-7)$: price at the i^{th} hour on the $(j-7)^{th}$ day,

- $Price(i-1, j-1)$: price at the $(i-1)^{th}$ hour on the $(j-1)^{th}$ day,
- $Price(i-2, j-1)$: price at the $(i-2)^{th}$ hour on the $(j-1)^{th}$ day,

In this proposed case study, the number of input variables was increased to six, as weekly information of the electricity price for that specific hour was also considered.

Figs. 6.9 and 6.10 illustrate the testing performances for minimum (4h) and maximum (22h) electricity price forecasting using case study 2, for all simulated forecasting models, while Tables 6.4 and 6.5 summarise their statistical performances.

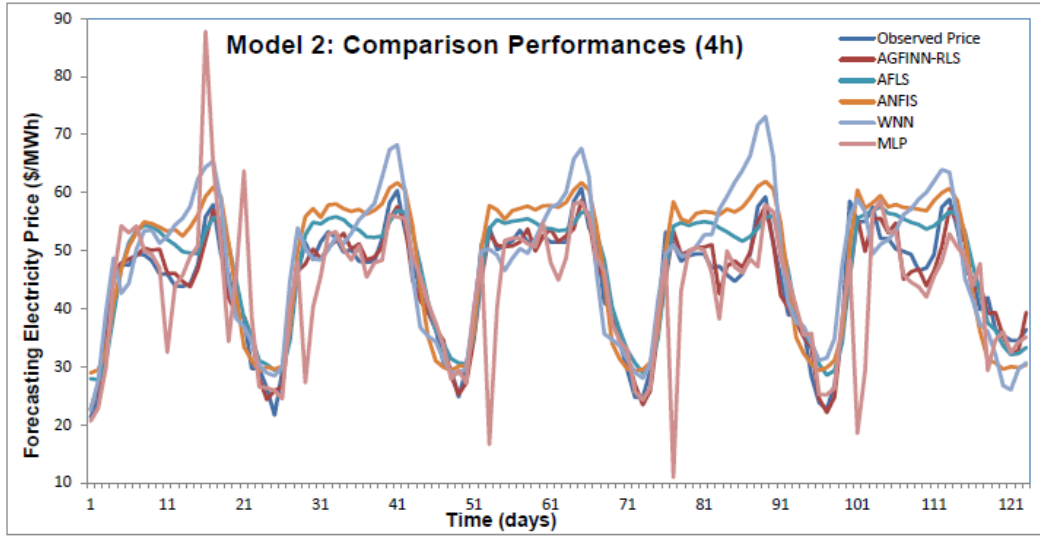


Fig. 6.9 Forecasting Electricity Price at 4h (Case Study 2)

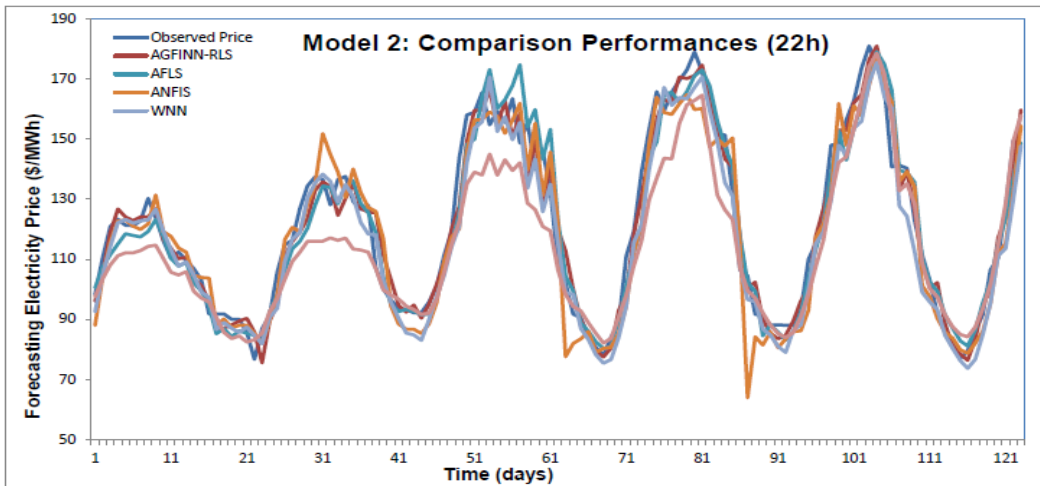


Fig. 6.10 Forecasting Electricity Price at 22h (Case Study 2)

Based on this configuration, AGFINN models have been involved in forecasting the maximum (22h) and minimum (4h) price respectively. AGFINN-RLS models produced best results again. From both forecasting graphs and statistical performances, it is obvious that

results have been improved compared to case study 1. This simulation and the improved results verified the assumption that electricity prices “contain” a periodicity effect.

| | AGFINN RLS | AGFINN TSK | AGFINN CA | AFLS | ANFIS | WNN | MLP |
|----------------------|-----------------------|-----------------------|----------------------|-------------|--------------|------------|------------|
| RMSE | 2.9220 | 3.3089 | 3.7748 | 4.5427 | 6.0034 | 6.7550 | 8.7796 |
| APE | 637.484 | 770.647 | 929.999 | 1160.1 | 1515.4 | 1493.2 | 1306.4 |
| MAPE | 5.1828 | 6.2654 | 7.5610 | 9.4320 | 12.3199 | 12.1400 | 10.6214 |
| MAE | 2.2178 | 2.6347 | 3.1964 | 3.9305 | 5.2979 | 5.2973 | 4.8608 |
| SEP | 6.5371 | 7.4027 | 8.4450 | 10.1630 | 13.4307 | 15.1124 | 19.6418 |
| U₁ | 0.0321 | 0.0361 | 0.0402 | 0.0484 | 0.0629 | 0.0710 | 0.0969 |
| R² | 0.9555 | 0.9453 | 0.9580 | 0.9214 | 0.9197 | 0.8729 | 0.6799 |

Table 6.4 Statistical performance for Electricity Price Forecasting Models at 4h

| | AGFINN RLS | AGFINN TSK | AGFINN CA | AFLS | ANFIS | WNN | MLP |
|----------------------|-----------------------|-----------------------|----------------------|-------------|--------------|------------|------------|
| RMSE | 6.9356 | 7.3320 | 7.6515 | 7.9278 | 9.8380 | 8.5303 | 11.6525 |
| APE | 540.905 | 562.889 | 560.288 | 591.165 | 759.612 | 666.332 | 855.803 |
| MAPE | 4.3976 | 4.5763 | 4.5552 | 4.8062 | 6.1757 | 5.4173 | 6.9578 |
| MAE | 5.2993 | 5.6935 | 5.7602 | 5.9638 | 7.4261 | 6.6674 | 9.0974 |
| SEP | 5.7317 | 6.0593 | 6.3233 | 6.5517 | 8.1302 | 7.0496 | 9.6298 |
| U₁ | 0.0280 | 0.0297 | 0.0311 | 0.0320 | 0.0400 | 0.0350 | 0.0484 |
| R² | 0.9703 | 0.9686 | 0.9660 | 0.9620 | 0.9462 | 0.9674 | 0.9552 |

Table 6.5 Statistical performances for Electricity Price Forecasting Models at 22h

The optimal structure for AGFINN-RLS included 20 fuzzy rules for both cases. As it is illustrated from these statistical tables, an AFLS NF model utilising 20 and 25 rules for 4h and 22h respectively, achieved a satisfactory performance, although inferior to AGFINN-CA model. Both AFLS and AGFINN-CA shared the same type of defuzzification scheme. Fig. 6.11 illustrates the performances of AGFINN-based models for this case study.

Both MLP and ANFIS forecasting results were problematic as they revealed a number of unexpected “spikes” and 4h and 22h respectively. The MLP NN retained the same network configuration as in case study 1, while in the case of ANFIS, this performance was achieved with a high computational cost, by utilizing two membership functions for each input variables and 64 fuzzy rules. In contrast, WNN model reveal a remarkable robustness against ANFIS, resulting in a similar performance with only 20 wavelet functions.

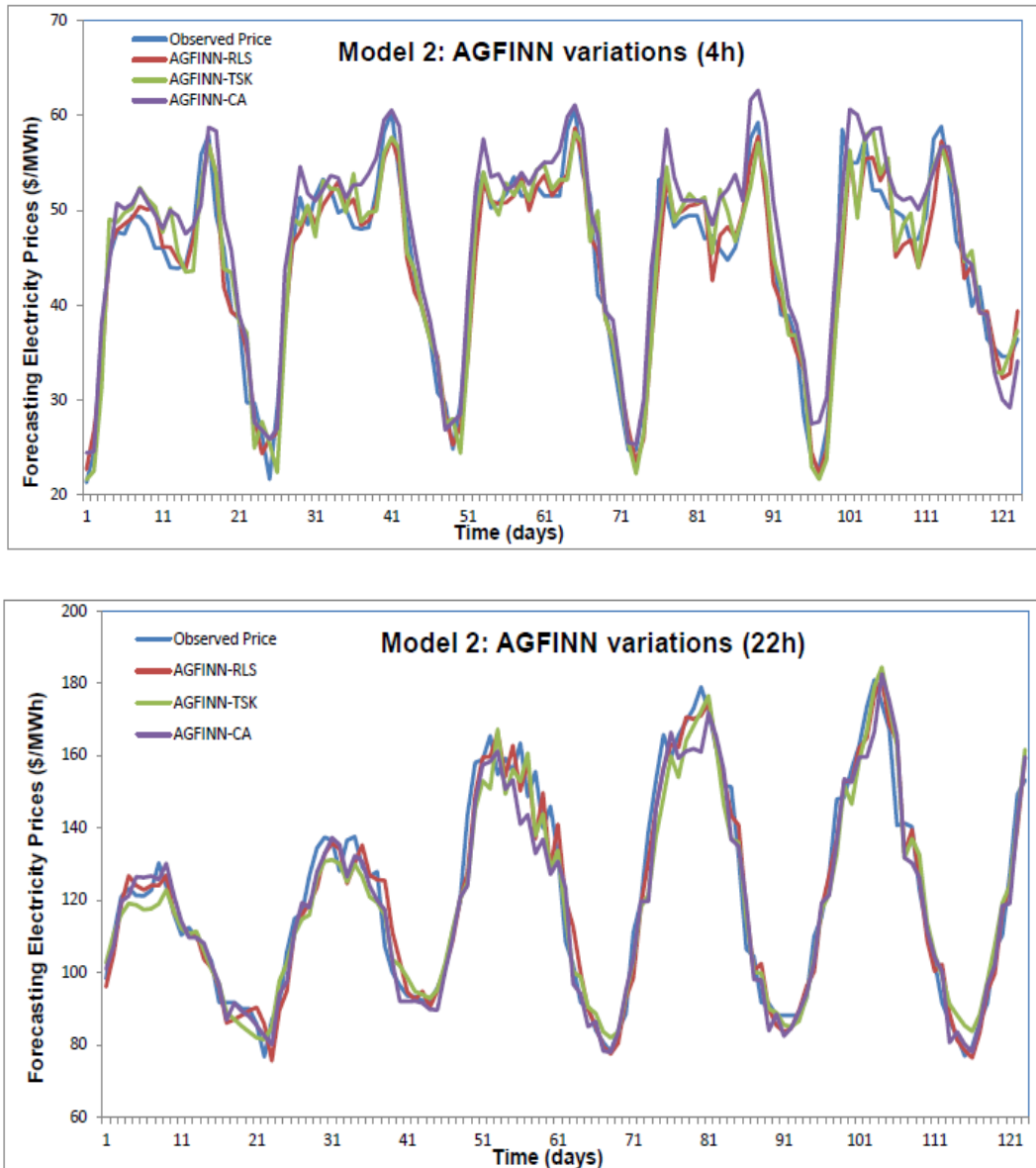


Fig. 6.11 Comparison of AGFINN-based models (Case Study 2)

6.4.3 Case Study 3

The objective of this model is to expand case study 2, by adding the exogenous input of the forecasted electricity load from case study 1. It was considered as very important to evaluate the combined effect of exogenous input together with the weekly periodicity characteristics of the price. Thus, for electricity price modelling for a specific hour (i) and day (j), the following seven input variables have been considered:

Target:

- $Price(i,j)$: electricity price at the i^{th} hour on the $(j)^{th}$ day,

Inputs:

- $Price(i, j-1)$: price at the i^{th} hour on the $(j-1)^{th}$ day,
- $Price(i, j-2)$: price at the i^{th} hour on the $(j-2)^{th}$ day,

- $Price(i, j-3)$: price at the i^{th} hour on the $(j-3)^{th}$ day,
- $Price(i, j-7)$: price at the i^{th} hour on the $(j-7)^{th}$ day,
- $Price(i-1, j-1)$: price at the $(i-1)^{th}$ hour on the $(j-1)^{th}$ day,
- $Price(i-2, j-1)$: price at the $(i-2)^{th}$ hour on the $(j-1)^{th}$ day,
- $Load(i, j)$: electricity load at the i^{th} hour on the j^{th} day,

In this case study, the number of input variables was increased to seven, due to the inclusion of the forecasted value of electricity load. Figs. 6.12 and 6.13 illustrate the testing performances for minimum (4h) and maximum (22h) electricity price forecasting using case study 3, for all simulated forecasting models.

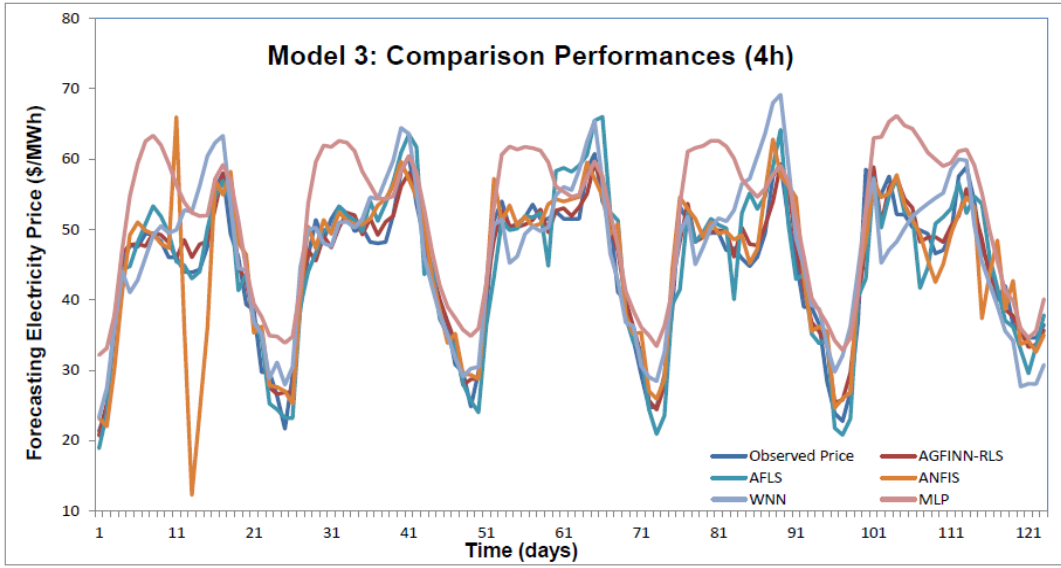


Fig. 6.12 Forecasting Electricity Price at 4h (Case Study 3)

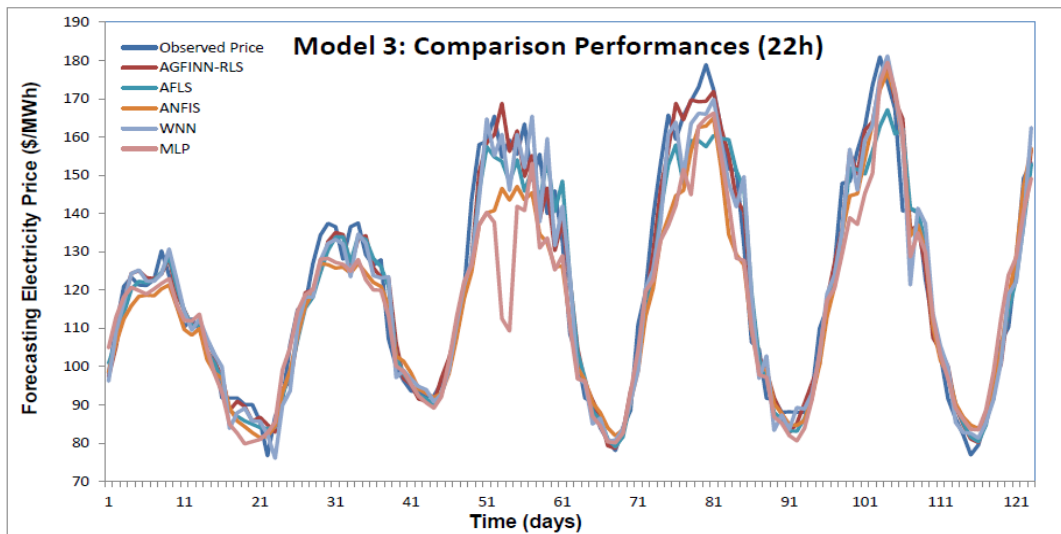


Fig. 6.13 Forecasting Electricity Price at 22h (Case Study 3)

As it is illustrated from Fig. 6.12, AGFINN, AFLS and WNN models managed to capture the dynamics (*i.e.* trend) at 4h with different levels of accuracy. For this experiment, AGFINN-RLS utilized 20 rules, while the optimal number of rules for AFLS was 25. WNN performed satisfactory with 20 Morlet wavelet functions. On the other hand, forecasting via ANFIS model resulted in unexpected spikes. This issue is due to the complexity of the specific ANFIS model. With seven input variables and two MFs per input, a massive number of 128 rules required for training purposes. ANFIS's performance was significantly improved for the case of 22h, as shown in Fig. 6.13. MLP's performance was deteriorated however for this case.

| | AGFINN RLS | AGFINN TSK | AGFINN CA | AFLS | ANFIS | WNN | MLP |
|----------------------|-----------------------|-----------------------|----------------------|-------------|--------------|------------|------------|
| RMSE | 2.6612 | 2.9988 | 3.5844 | 4.3667 | 5.4409 | 5.4184 | 8.0055 |
| APE | 608.757 | 674.439 | 857.814 | 944.293 | 1023 | 1273.9 | 2040.3 |
| MAPE | 4.9492 | 5.4832 | 6.9741 | 7.6772 | 8.3168 | 10.3565 | 16.5878 |
| MAE | 2.0810 | 2.2845 | 2.9241 | 3.34409 | 3.5089 | 4.3761 | 6.8644 |
| SEP | 5.9536 | 6.7089 | 8.0189 | 9.7692 | 12.1724 | 12.1219 | 17.9098 |
| U₁ | 0.0290 | 0.0326 | 0.0383 | 0.0474 | 0.0590 | 0.0579 | 0.0815 |
| R² | 0.9635 | 0.9555 | 0.9588 | 0.9229 | 0.8630 | 0.8802 | 0.9079 |

Table 6.6 Statistical performance for Electricity Price Forecasting Models at 4h

| | AGFINN RLS | AGFINN TSK | AGFINN CA | AFLS | ANFIS | WNN | MLP |
|----------------------|-----------------------|-----------------------|----------------------|-------------|--------------|------------|------------|
| RMSE | 6.4605 | 6.8514 | 7.5032 | 7.7340 | 9.1584 | 8.0368 | 11.4835 |
| APE | 487.999 | 521.740 | 616.196 | 550.734 | 655.688 | 606.248 | 739.416 |
| MAPE | 3.9675 | 4.2418 | 5.0097 | 4.4775 | 5.3308 | 4.9288 | 6.0115 |
| MAE | 4.8768 | 5.2331 | 5.9290 | 5.7301 | 6.9329 | 6.1892 | 7.9360 |
| SEP | 5.3391 | 5.6621 | 6.2007 | 6.3915 | 7.5686 | 6.6417 | 9.4901 |
| U₁ | 0.0260 | 0.0277 | 0.0303 | 0.0314 | 0.0377 | 0.0325 | 0.0473 |
| R² | 0.9742 | 0.9718 | 0.9699 | 0.9670 | 0.9682 | 0.9604 | 0.9388 |

Table 6.7 Statistical performance for Electricity Price Forecasting Models at 22h

Tables 6.6 and 6.7 summarise statistical performances for minimum (4h) and maximum (22h) electricity price forecasting using case study 3, for all simulated forecasting models. AGFINN models in both experiments outperformed “rival” models, as shown also from the statistical tables.

Although, case study 3 proved to be the most accurate compared to previous cases, even in this case, there were variations in performance among AGFINN-based models. All models utilise FCM clustering units, as well as asymmetric functions as MFs.

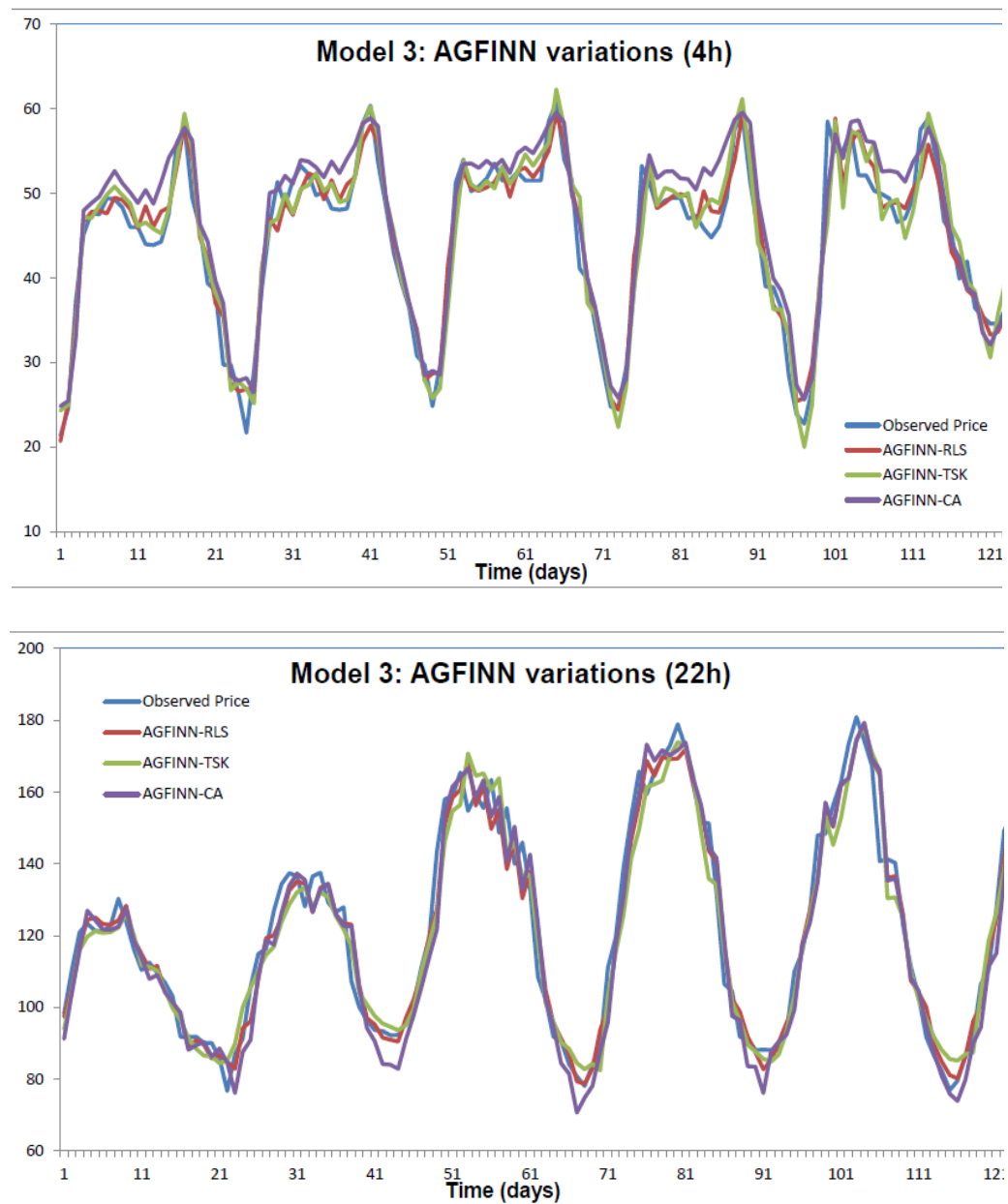


Fig. 6.14 Comparison of AGFINN-based models (Case Study 3)

The difference in defuzzification part is mainly responsible for such variations. It was verified that TSK-based models are more accurate/suitable for forecasting applications that CA-based systems. AGFINN-RLS utilised the recursive least squares for tuning the linear parameters at TSK part, similarly to the case of ANFIS. Fig. 6.14 illustrates the performances of AGFINN-based models for this case study.

The plot of predicted (via AGFINN-RLS) versus observed electricity prices is illustrated in Fig. 6.15, and shows a very good distribution around the line of equity ($y=x$). This is also verified by the R^2 index which indicated a very good fit of the observed data from the AGFINN-RLS based approach for both 4h and 22h tests.

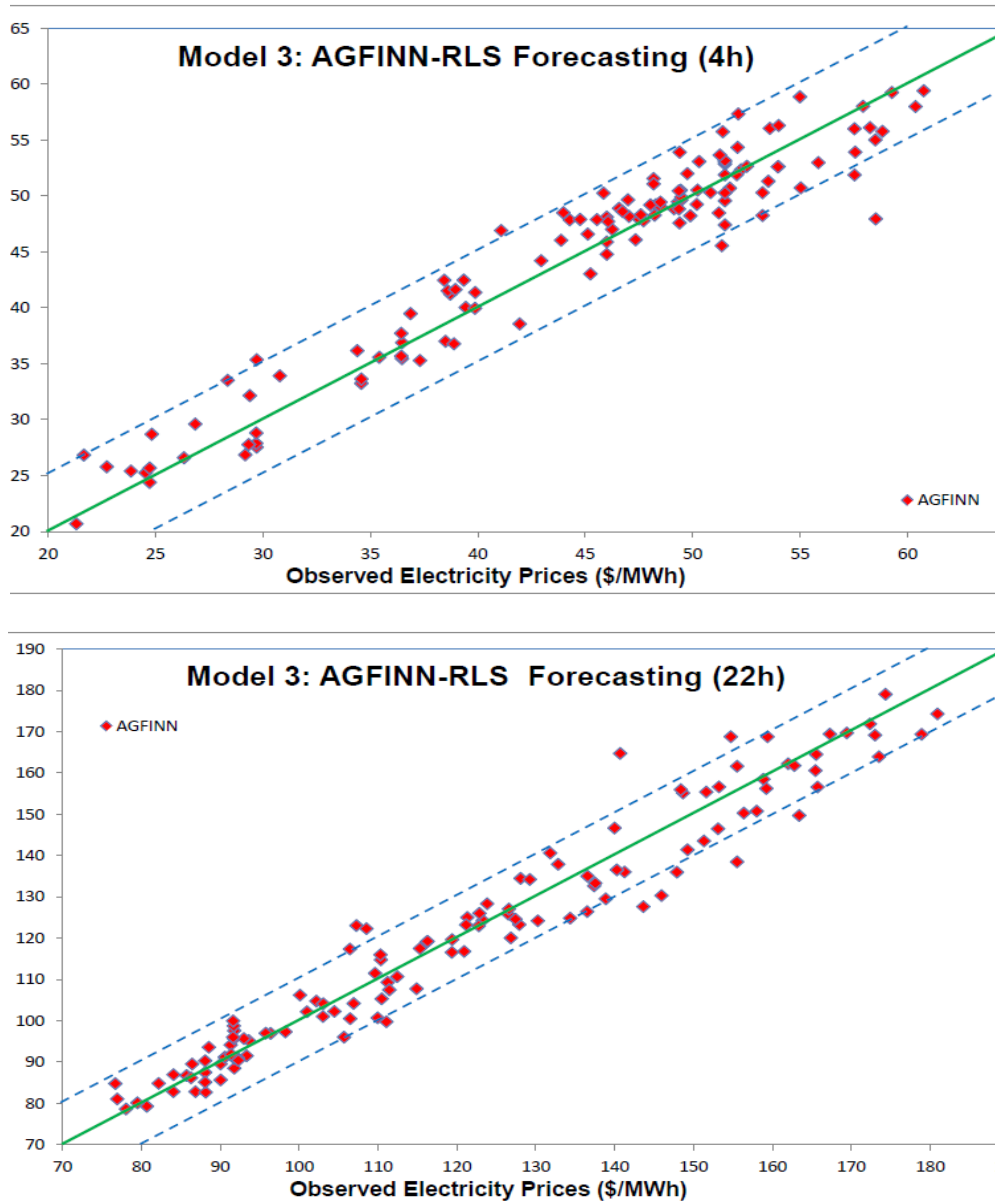


Fig. 6.15 AGFINN-RLS Forecasting performance (Case Study 3)

Electricity price forecasting has become essential to power consumers and producers in the deregulated electricity market. Developing an effective and accurate forecasting model has thus become a very important task. This work presented an hourly-ahead short-term electricity price forecast by using novel neurofuzzy modelling approaches in ISO New England market. In ISO New England market, the main challenging issue is that the hourly

market price curves are highly volatile. AGFINN models utilised Fuzzy C-Means clustering method for identifying the number of MFs/rules, while asymmetric MFs have replaced the traditional Gaussian functions in the fuzzification component of the NF architecture. Two distinct approaches have been investigated based on the defuzzification scheme. The effectiveness of the proposed approaches has been thoroughly assessed by comparing them with alternative neural or neurofuzzy techniques, via three case studies. Future research includes the incorporation in the modelling process additional exogenous parameters, as well as the adoption of an alternative clustering method for the pre-processing stage. There is need to explore further the use of hybrid intelligent systems in the area of electricity price forecasting, and this research has contributed towards this goal.

Chapter Seven

CONCLUSIONS AND FUTURE ENHANCEMENTS

7.1 Conclusions

Nearly two decades back nonlinear system identification consisted of several ad-hoc approaches, which were restricted to a very limited class of systems. However, with the advent of the various soft computing methodologies like neural networks and the fuzzy logic combined with modern structure optimization techniques, a wider class of systems can be handled at present. Complex systems may be of diverse characteristics and nature. These systems may be linear or nonlinear, continuous or discrete, time varying or time invariant, static or dynamic, short term or long term, central or distributed, predictable or unpredictable, ill or well defined. System outputs may be measurable or not measurable. Models of real systems are of fundamental importance in virtually all disciplines and hence there is a strong demand for advanced modelling and identification schemes. This is because models help in system's analysis, which in turn helps to get a better understanding of the system for predicting or simulating a system's behaviour. The challenges involved in modelling and identification of a nonlinear system are too many and efforts had been made to tackle them by applying various soft computing methodologies. Neural networks and fuzzy systems have been combined to join their advantages and to cure their individual "illnesses". Neural networks introduced their computational characteristics of learning in the fuzzy systems and received from them the interpretation and clarity of systems representation. Thus, the disadvantages of the fuzzy systems are compensated by the capacities of the neural networks. Throughout main parts of this thesis, an attempt has been made to construct advanced neurofuzzy (NF) architectures and verify their capabilities through a number of real case studies. Obviously, during this process, extensive comparison against established modelling schemes has been considered in order to justify the adoption/use of the proposed methods. Reviewing chapters 4, 5 and 6, we could easily identify the "evolution" of the proposed attempt to create a novel multipurpose neurofuzzy modelling architecture.

In chapter 4, an Adaptive Fuzzy Logic System (AFLS) model was implemented to associate FTIR spectral data with beef spoilage. According to literature, for the first time in the world, neurofuzzy systems have been used in this specific area. The AFLS model, a classic MIMO scheme, included a prototype defuzzification scheme and was different from conventional

fuzzy rule-table approaches that utilise the “look-up table” concept. In those models, as one fuzzy rule is normally assigned for each one of these subspaces, their main drawback is that the number of fuzzy rules increases exponentially with respect to the number of inputs n . The Adaptive NeuroFuzzy Inference System (ANFIS) is a classical example of such approach, where the number of fuzzy rules is related to the number of input variables as well as the number of membership functions for each input. In the case of AFLS, the number of memberships for each input variable is directly associated to the number of rules, hence, the “*curse of dimensionality*” problem is significantly reduced. The MIMO characteristics and the “fuzzification” structure of AFLS have been considered as important issues to be incorporated in the development of an advanced neurofuzzy model. Performance of AFLS was superior to that of ANFIS, while prediction performances of MLP and PLS schemes revealed the deficiencies of these systems, which have been used extensively in the area of Food Microbiology.

The methodology proposed in Chapter 5, addresses some interesting ideas that could be incorporated in an advanced NF scheme. The main objective of that chapter was to associate, for the first time again according to literature, spectral data acquired by multispectral imaging techniques with meat spoilage, using neurofuzzy systems. In addition, for the first time, an identification model was built to predict the temperatures under which meat samples were stored. The modelling aspects of this case study are associated with the concept of MISO modelling schemes. It is well known that TSK-based NFs, such as ANFIS, have enjoyed a great success in modelling single output nonlinear problems. The realization of AFINN model follows also the classic TSK structure, incorporating however a clustering unit in the fuzzification section and an additional internal competitive clustering layer. Unlike the ANFIS system, where the number of local linear systems is the same as the number of rules, AFINN provides a means of controlling the growth of the number of local linear systems when the order of the system under consideration increases, so that least-squares estimation can be applied without performance degradation. A clustering algorithm is applied for the sample data in order to organize feature vectors into clusters, such that points within a cluster are closer to each other than vectors belonging to different clusters. The fuzzy rule base is derived using results obtained from a clustering algorithm. Results from AFINN scheme revealed its superiority, against models based on ANFIS, multilayer neural networks (MLP), Non-linear regression and PLS schemes.

The knowledge gained from the previous two case studies were summarised to the following desirable objectives for an advanced NF scheme:

- A NF model must have minimum possible number of rules
- A NF model must be generic acting either as MISO or MIMO identification model

- A NF model must have a versatile nonlinear membership function.

Based on these objectives, the AGFINN architecture was developed in Chapter 6 and was evaluated to an electricity price forecasting problem. AGFINN models utilised fuzzy C-means clustering method for identifying the number of MFs/rules, while asymmetric MFs have replaced the traditional Gaussian functions in the fuzzification component of the NF architecture. AGFINN has been designed acting either as an MISO or MIMO system. In the MISO case, a TSK defuzzification scheme has been adopted, while for the MIMO case, a “centre average” (CA) defuzzifier has been used as defuzzification scheme. For the case of TSK scheme, a hybrid learning algorithm has been also developed. The proposed modelling schemes are compared successfully against AFLS, ANFIS, Wavelet network (WNN) and MLP NN modelling schemes in order to assess their prediction performance.

7.2 Future Enhancements

There are a number of issues that have been left out of the scope in the present thesis, which could be considered as a future work. Future enhancements of this research’s field can be aligned in several directions. Points mentioned below, are just some of the possible extensions that could be done as future work.

Advanced Clustering Algorithm: Although the hypothesis of having a pre-processing clustering step proved to be valid for the AGFINN scheme, there is need for a more efficient unsupervised clustering algorithm, where the number of clusters needs to be identified from the algorithm itself. For real applications, we might look also for a “topology”- based algorithms, such as growing cells or growing neural gas.

Asymmetric Function: The hypothesis of having an asymmetric fuzzy membership function proved to be valid. It could be interesting to see the incorporation of asymmetric function to wavelet functions used as MFs in fuzzy wavelet models. Alternatively, AGFINN could be transformed to a fuzzy wavelet version by replacing the polynomial function at the TSK defuzzification with a wavelet NN.

Adaptive structure/parameter learning: Off-line clustering methods require that data should be ready before the modelling. Obviously, it is difficult for human experts to examine all the input–output data from a real complex system to find a number of proper rules for the fuzzy system. Hence, an alternative way is to consider a two learning stages procedure, the structure learning phase and the parameter learning phase. These two phases are done simultaneously. Rules and parameters are created and adapted as on-line learning proceeds via simultaneous structure and parameter identification. Extended Kalman Filter as a learning algorithm could also replace the classic GD algorithm.

REFERENCES

1. S. J. Russell and P. Norvig, *Artificial intelligence a modern approach*, Pearson, 2016.
2. R. Isermann and M. Munchhof, *Identification of Dynamic Systems An Introduction with Applications*, Springer London, 2011.
3. L. Ljung, *System Identification Toolbox™ User's Guide*, (R2016b), 2016.
4. A. K. Tangirala, *Principles of System Identification, Theory and Practice*, 2015.
5. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*, 4th ed. John Wiley & Sons, 2007.
6. C. Chatfield, *The analysis of time series: an introduction*, 6th ed. WILEY, 2004.
7. S. Ho, M. Xie, and T. Goh, "A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction," *Comput. Ind. Eng.*, vol. 42, no. 2–4, pp. 371–375, Apr. 2002.
8. S. J. Shyh-Jier Huang and K. R. Kuang-Rong Shih, "Short-term load forecasting via ARMA model identification including non-gaussian process considerations," *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 673–679, May 2003.
9. G. Acuna, C. Ramirez, and M. Curilem, "Comparing NARX and NARMAX models using ANN and SVM for cash demand forecasting for ATM," *The International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2012.
10. S. M. Biyiksiz, "ARMA modeling based on partial AR and MA parameter approximation," *Spectrum Estimation and Modeling*, Fourth Annual ASSP Workshop on, Minneapolis, MN USA, pp. 397–401, 1988.
11. O. Nelles, *Nonlinear System Identification, From Classical Approaches to Neural Networks and Fuzzy Models*, Springer, Verlag Berlin Heidelberg, 2001.
12. L. A. Zadeh, "Fuzzy sets, Information and Control," vol. 8, pp. 338–353, 1965.
13. C.-T. Lin and C. S. G. Lee, *Neural Fuzzy Systems*, New Jersey, Prentice - Hall, 1996.
14. T. J. Ross, *Fuzzy Logic with Engineering Applications*, Third. John Wiley & Sons Ltd, 2009.
15. L. Fausett, *Fundamentals of neural networks*, edition 1. vol. 1. New Jersey, Prentice Hall, 1994.
16. F. M. Ham and I. Kostanic, *Principles of Neurocomputing for science and engineering*, New York, McGraw Hill, 2001.
17. Y. Li and W. Ma, "Applications of Artificial Neural Networks in Financial Economics: A Survey," *International Symposium on Computational Intelligence and Design*, pp. 211–214, 2010.
18. M. Layouni, S. Tahar, and M. S. Hamdi, "A survey on the application of Neural Networks in the safety assessment of oil and gas pipelines," *IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES)*, pp. 95–102, 2014.
19. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy Of Sciences of the USA*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.
20. T. Kohonen, *Self-organising maps*, Springer, Berlin, 1995.

21. L. H. Tsoukalas, and R. E. Uhrig, *Fuzzy and neural approaches in engineering*, John Wiley & Sons, Inc. New York, NY, USA, 1996.
22. J. S. R. Jang, C. T. Sun and E. Mizutani, *Neuro fuzzy and Soft Computing: A Computation Approach to Learning and Machine Intelligence*, Pearson Education USA, 6 Oct. 1996.
23. J. H. Holland, *Adaptation in Natural and Artificial Systems*, Cambridge, MA, MIT Press. Second edition, 1992.
24. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA, Addison-Wesley, 1989.
25. X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, pp. 1423-1447, 1999.
26. F. Herrera, M. Lozano, and J. L. Verdegay, "Tuning fuzzy logic controllers by genetic algorithms* 1," *International Journal of Approximate Reasoning*, vol. 12, pp. 299-315, 1995.
27. O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, and L. Magdalena, "Ten years of genetic fuzzy systems: current framework and new trends," *Fuzzy Sets and Systems*, vol. 141, pp. 5-31, 2004.
28. L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans on Systems, Man, and Cybernetics*, vol. 3, no. 1, pp. 28-44, Jan. 1973.
29. J. H. Lilly, *Fuzzy Control and Identification*, John Wiley, 2010.
30. R.-E. Precup and H. Hellendoorn, "A survey on industrial applications of fuzzy control," *Comput. Ind.*, vol. 62, no. 3, pp. 213-226, Apr. 2011.
31. M. Mahfouf, M. F. Abbod, and D. A. Linkens, "A survey of fuzzy logic monitoring and control utilisation in medicine," *Artif. Intell. Med.*, vol. 21, pp. 27-42, 2001.
32. O. F. Bay, and A. B. Usakli, "Survey of Fuzzy Logic Applications in Brain-Related Researches," *J. Med. Syst.*, vol. 27, no. 2, pp. 215-223, 2003.
33. N. Bharathan, "A Survey on the Applications of Fuzzy Logic in Medical Diagnosis," *Int. J. Sci. Eng. Res.*, vol. 4, no. 4, 2013.
34. A. Mudassir, S. Akhtar, H. Kamel, and N. Javaid, "A Survey on Fuzzy Logic Applications in Wireless and Mobile Communication for LTE Networks," *10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*, pp. 76-82, 2016.
35. H. J. Zimmermann, *Fuzzy Set Theory & its Applications*, Kluwer Academic Press, Boston, MA, 2nd edition, 1993.
36. E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Mach. Stud.*, vol. 7, pp. 1-13, 1975.
37. T. Takagi, and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116-132, 1985.
38. M. Sugeno, and G. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, no. 1, pp. 15-33, 1988.

39. E. Kim, H. Lee, M. Park, "A simple identified Sugeno-type fuzzy model via double clustering," *Inform. Sci.*, vol. 110, pp. 25-39, 1998.
40. G. W. Irwin, K. Warwick, and K. J. Hunt, *Neural Network Applications in Control*, The Institution of Electrical Engineers, 1995.
41. S. Haykin, *Neural Networks: A comprehensive Foundation*, Upper saddle River, NJ, Prentice Hall. 1994.
42. Z. Shi, L. He, K. Suzuki, T. Nakamura, and H. Itoh, "Survey on Neural Networks Used for Medical Image Processing.," *Int. J. Comput. Sci.*, vol. 3, no. 1, pp. 86–100, Feb. 2009.
43. F. N. Chowdhury, P. Wahi, R. Raina, and S. Kaminedi, "A survey of neural networks applications in automatic control," *Proceedings of the 33rd Southeastern Symposium on System Theory (Cat. No.01EX460)*, pp. 349–353, 2001.
44. H.-C. ZHANG and S. H. HUANG, "Applications of neural networks in manufacturing: a state-of-the-art survey," *Int. J. Prod. Res.*, vol. 33, no. 3, pp. 705–728, Mar. 1995.
45. W. I. Schollhorn and J. M. Jager, "A Survey on Various Applications of Artificial Neural Networks in Selected Fields of Healthcare," *Neural Networks in Healthcare*, IGI Global, pp. 20–59, 2006.
46. Y. Li and W. Ma, "Applications of Artificial Neural Networks in Financial Economics: A Survey," *2010 International Symposium on Computational Intelligence and Design*, pp. 211–214, 2010.
47. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
48. V. B. Daohang Sha, "An on-line hybrid learning algorithm for multilayer perceptron in identification problems," *Elsevier Computers and Electrical Engineering*, 2002.
49. R. K. Young, *Wavelet theory and its applications*, Kluwer Academic Publishers, 1993.
50. J. Zhang, and G. Gilbert, "Wavelet Neural Networks for Function Learning," *IEEE Trans. Signal Processing*, vol. 2, no. 6, pp. 1485-1496, 1995.
51. A. B. Stephen, and H. L. Wei, "A new class of wavelet networks for nonlinear system identification," *IEEE Trans Neural Netw*, vol. 16, no. 4, pp. 862–874, 2005.
52. B. Kosko, *Neural networks and fuzzy systems: a dynamical systems approach*, Prentice Hall, Englewood Cliffs, NJ, 1991.
53. A. Saad, "An Overview of Hybrid Soft Computing Techniques for Classifier Design and Feature Selection," in *Hybrid Intelligent Systems, 2008. HIS '08. Eighth International Conference on*, pp. 579-583, 2008.
54. H. R. Berenji, "A reinforcement learning—based architecture for fuzzy logic control," *Int. J. Approx. Reason.*, vol. 6, no. 2, pp. 267–292, Feb. 1992.
55. D. Nauck, F. Klawonn, and R. Kruse, *Foundations of neuro-fuzzy systems*, John Wiley & Sons, Inc. New York, NY, USA, 1997.

56. N. Walia, S. Kumar, and H. Singh, "A Survey on Applications of Adaptive Neuro Fuzzy Inference System," *Int. J. Hybrid Inf. Technol.*, vol. 8, no. 11, pp. 343–350, 2015.
57. V. S. Kodogiannis, and A. Lolis, "Prediction of foreign exchange rates by neural network and fuzzy system based techniques," *European Symposium on Artificial Neural Networks*, Bruges, Belgium, 2001.
58. J. S. R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference System," *IEEE Trans. On System, Man and Cybernetics*, vol. 23, no. 3, pp. 665-685, 1993.
59. H. Yang, J. Li, and F. Ding, "A neural network learning algorithm of chemical process modeling based on the extended Kalman filter," *Neurocomputing*, vol. 70, pp. 625-632, 2007.
60. W. Pedrycz, *Knowledge-Based Clustering*. Hoboken, NJ, USA, John Wiley & Sons, Inc., 2005.
61. D. Xu, and Y. Tian, "A Comprehensive Survey of Clustering Algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.
62. R. Duda, and P. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
63. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
64. N. R. Pal, and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *Fuzzy Systems, IEEE Transactions on*, vol. 3, pp. 370-379, 1995.
65. A. Baraldi, and P. Blonda, "A survey of Fuzzy Clustering Algorithms for pattern Recognition Part II," *IEEE Trans on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 6, pp. 786-801, 1999.
66. E. M. Desmond, T. A. Kenny, P. Ward, and D. W. Sun, "Effect of rapid and conventional cooling methods on the quality of cooked ham joints," *Meat Science*, vol. 56, no. 3, pp. 271–277, 2000.
67. G. ElMasry, D. W. Sun, "Meat Quality Assessment Using a Hyperspectral Imaging System.," *In: Hyperspectral Imaging for Food Quality Analysis and Control* (Edited by D.W. Sun), pp. 273 -294, 2010.
68. A. M. Herrero, P. Carmona, I. Lopez-Lopez, and F. Jimenez-Colmenero, "Raman spectroscopic evaluation of meat batter structural changes induced by thermal treatment and salt addition," *Journal of Agricultural and Food Chemistry*, vol. 56, no. 16, pp. 7119–7124, 2008.
69. D. I. Ellis, and R. Goodacre, "Rapid and quantitative detection of the microbial spoilage of muscle foods: current status and future trends," *Trends in Food Science & Technology*, vol. 12, pp. 414-424, 2001.
70. B. Dissing, O. Papadopoulou, C. Tassou, B. Ersbøll, J. Carstensen, E. Panagou, and G-J.E. Nychas, "Using Multispectral Imaging for Spoilage Detection of Pork Meat," *Food and Bioprocess Technology*, vol. 6, no. 9, pp. 2268-2279, 2013.

71. S. Balasubramanian, S. Panigrahi, C. M. Logue, H. Gu, and M. Marchello, "Neural networks-integrated metal oxide-based artificial olfactory system for meat spoilage identification," *J. Food Eng.*, vol. 91, pp. 91-98, 2009.
72. D. J. M. Mouwen, A. Hörman, H. Korkeala, A. Alvarez-Ordóñez, and M. Prieto, "Applying Fourier-transform infrared spectroscopy and chemometrics to the characterization and identification of lactic acid bacteria," *Vibrational Spectroscopy*, vol. 56, no. 2, pp. 193-201, 2011.
73. X. Tian, J. Wang, and S. Cui, "Analysis of pork adulteration in minced mutton using electronic nose of metal oxide sensors," *Journal of Food Engineering*, vol. 119, no. 4, pp. 744-749, 2013.
74. G.-J.E. Nychas, P. N. Skandamis, C. C. Tassou, and K. P. Koutsoumanis, "Meat spoilage during distribution," *Meat Science*, vol. 78, pp. 77-89, 2008.
75. J. N. Miller, and J. C. Miller, *Statistics and Chemometrics for Analytical Chemistry*, Prentice Hall, 2005.
76. A. Guillén, F. G. del Moral, L. J. Herrera, G. Rubio, I. Rojas, O. Valenzuela, and H. Pomares, "Using near-infrared spectroscopy in the classification of white and Iberian pork with neural networks," *Neural Computing Applications*, vol. 19, pp. 465-470, 2010.
77. J. Qin, K. Chao, and M. S. Kim, "Investigation of Raman chemical imaging for detection of lycopene changes in tomatoes during postharvest ripening," *Journal of Food Engineering*, vol. 107, no. 3/4, pp. 277-288, 2011.
78. E. Z. Panagou, and V. Kodogiannis, "Application of Neural Networks as a Non-linear Modelling Technique in Food Mycology," *Expert Systems with Applications*, vol. 36, pp. 121-131, 2009.
79. D. Rutkowska, *Neuro-Fuzzy Architectures and Hybrid Learning*, Springer, 2002.
80. O. Nelles, *Nonlinear System Identification*, Springer, Berlin, 2000.
81. SYMBIOSIS-EU project, Appendix 1, 2009. Available from: <http://www.aua.gr/tmhmeta/biomhxan/nychas/SYMBIOSIS12m/SYMBIOSIS1stYSuBRv070110Appendix1.pdf>.
82. A. A. Argyri, E. Z. Panagou, P. A. Tarantilis, M. Polysiou, and G.-J.E. Nychas, "Rapid qualitative and quantitative detection of beef fillets spoilage based on Fourier transform infrared spectroscopy data and artificial neural networks," *Sensors and Actuators B*, vol. 145, pp. 146-154, 2010.
83. R. Karoui, and J. De Baerdemaeker, "A review of the analytical methods coupled with chemometric tools for the determination of the quality and identity of dairy products," *Food Chemistry*, vol. 102, no. 3, pp. 621-640, 2007.
84. E. F. Olsen, E. O. Rukke, B. Egelandsdal, and T. Isaksson, "Determination of omega-6 and omega-3 fatty acids in pork adipose tissue with nondestructive Raman and fourier transform infrared spectroscopy," *Applied Spectroscopy*, vol. 62, no. 9, pp. 968-974, 2008.
85. C. Kirschner, R. Ofstad, H. J. Skarpeid, V. Host, and A. Kohler, "Monitoring of denaturation processes in aged beef loin by Fourier transform infrared

- microspectroscopy,” *Journal of Agricultural and Food Chemistry*, vol. 52, no. 12, pp. 3920–3929, 2004.
86. Z. Wu, H. C. Bertram, A. Kohler, U. Bocker, R. Ofstad, and H. J. Andersen, “Influence of aging and salting on protein secondary structures and water distribution in uncooked and cooked pork. A combined FT-IR microspectroscopy and H NMR study,” *Journal of Agricultural and Food Chemistry*, vol. 54, pp. 8589-8597, 2006.
 87. A. Rohman, Y. Sismindari, D. Erwanto, B. Yaakob, and B. Che Man, “Analysis of pork adulteration in beef meatball using Fourier transform infrared (FTIR) spectroscopy,” *Meat Science*, vol. 88, pp. 91–95, 2011.
 88. N. Vlachos, Y. Skopelitis, M. Psaroudaki, V. Konstantinidou, A. Chatzilazarou, and E. Tegou, “Applications of Fourier-transform infrared spectroscopy to edible oils,” *Analysis Chimica Acta*, pp. 459-465, 2006.
 89. J. K. Amamcharla, S. Panigrahi, C. M. Logue, M. Marchello, and J. S. Sherwood, “Fourier transform infrared spectroscopy (FTIR) as a tool for discriminating *Salmonella typhimurium* contaminated beef,” *Sensing and Instrumentation for Food Quality and Safety*, vol. 4, no. 1, pp. 1-12, 2010.
 90. R. R. Gangidi, A. Proctor, and F. W. Pohlman, “Rapid Determination of Spinal Cord Content in Ground Beef by Attenuated Total Reflectance Fourier Transform Infrared Spectroscopy,” *Journal of Food Science*, vol. 68, no. 1, pp. 124-127, 2003.
 91. D. I. Ellis, D. Broadhurst, D. B. Kell, J. J. Rowland, and R. Goodacre, “Rapid and quantitative detection of the microbial spoilage of meat by Fourier transform infrared spectroscopy and machine learning” *Appl. Environ. Microbiol.*, vol. 68, pp. 2822–2828, 2002.
 92. Q. Chen, J. Cai, X. Wan, and J. Zhao, “Application of linear/non-linear classification algorithms in discrimination of pork storage time using Fourier transform near infrared (FT-NIR) spectroscopy,” *LWT - Food Science and Technology*, vol. 44, no. 10, pp. 2053-2058, 2011.
 93. O. Papadopoulou, E. Z. Panagou, C. C. Tassou, and G.-J.E. Nychas, “Contribution of Fourier transform infrared (FTIR) spectroscopy data on the quantitative determination of minced pork meat spoilage,” *Food Research International*, vol. 44, no. 10, pp. 3264-3271, 2011.
 94. A. Sahar, and E. Dufour, “Use of Fourier transform-infrared spectroscopy to predict spoilage bacteria on aerobically stored chicken breast fillets,” *LWT - Food Science and Technology*, vol. 56, pp. 315-320, 2014.
 95. J. Baranyi, and T. A. Roberts, “A dynamic approach to predicting bacterial growth in food,” *Int. J. Food Microbiol.*, vol. 23, pp. 277–294, 1994.
 96. V. S. Kodogiannis, and E. M. Anagnostakis, “Soft computing based techniques for short-term load forecasting,” *Fuzzy sets and systems*, vol. 128, no. 3, pp. 413-426, 2002.
 97. J. M. Mendel, “Fuzzy Logic Systems for Engineering: A Tutorial,” *Proceeding of The IEEE*, vol. 83, no. 3, pp. 345-377, 1995.

98. V. S. Kodogiannis, I. Petrounias and J.N. Lygouras, "Intelligent Classification using Adaptive Fuzzy Logic Systems," *IS2008 – IEEE International Conference on Intelligent Systems*, Bulgaria, 2008, pp. 8-13.
99. B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, 1992.
100. H. Yalcin, I. Ozturk, S. Karaman, O. Kisi, O. Sagdic, and A. Kayacier, "Prediction of Effect of Natural Antioxidant Compounds on Hazelnut Oil Oxidation by Adaptive Neuro-Fuzzy Inference System and Artificial Neural Network," *Journal of Food Science*, vol. 76, no. 4, pp. 112-120, 2011.
101. E. Z. Panagou, V. Kodogiannis, and G.-J.E. Nychas, "Modelling fungal growth using radial basis function neural networks: The case of the ascomycetous fungus *Monascus ruber* van Tieghem," *Int. Journal of Food Microbiology*, vol. 117, pp. 276-286, 2007.
102. E. W. Ross, I. A. Taub, C. J. Doona, F. E. Feeherry, and K. Kustin, "The mathematical properties of the quasi-chemical model for microorganism growth-death kinetics in foods," *International Journal of Food Microbiology*, vol. 99, pp. 157-171, 2005.
103. M. Amina, V. S. Kodogiannis, I. Petrounias, J. N. Lygouras and G.-J.E. Nychas, "Identification of the *Listeria Monocytogenes* survival curves in UHT whole milk utilising Local Linear Wavelet Neural Networks," *Expert Systems and Applications*, vol. 39, no. 1, pp. 1435-1450, 2012.
104. J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*, Prentice-Hall, 1997.
105. K. P. Singh, P. Ojha, A. Malik, and G. Jain, "Partial least squares and artificial neural networks modeling for predicting chlorophenol removal from aqueous solution," *Chemom. Intell. Lab. Syst.*, vol. 99, pp. 150-160, 2009.
106. D. S. Lee, M. W. Lee, S. H. Woo, Y.-J. Kim, and J. M. Park, "Nonlinear dynamic partial least squares modeling of a full-scale biological wastewater treatment plant," *Process Biochemistry*, vol. 41, no. 9, pp. 2050-2057, 2006.
107. M. Amina, E. Z. Panagou, V. S. Kodogiannis and G.-J.E. Nychas, "Wavelet Neural Networks for modelling high pressure inactivation kinetics of *Listeria monocytogenes* in UHT whole milk," *Chemometrics and intelligent laboratory systems*, vol. 103 no. 2, pp. 170-183, 2010.
108. J. L. Tan, "Meat quality evaluation by computer vision," *J. Food Eng.*, vol. 61, pp. 27-35, 2004.
109. H. Huang, L. Liu, and M. O. Ngadi, "Recent Developments in Hyperspectral Imaging for Assessment of Food Quality and Safety," *Sensors*, vol. 14, pp. 7248-7276, 2014.
110. N. Prieto, R. Roehe, P. Lavín, G. Batten, and S. Andrés, "Application of near infrared reflectance spectroscopy to predict meat and meat products quality: A review" *Meat Sci.* vol. 83, pp. 175-186, 2009.
111. V. S. Kodogiannis, and I. Petrounias, "Modelling of survival curves in food microbiology using adaptive fuzzy inference neural networks," *IEEE Int. Conf. on*

- Computational Intelligence for Measurement Systems and Applications*, CIMSAs, pp. 35-40, 2012.
112. Y. Z. Feng, and D. W. Sun, "Application of hyperspectral imaging in food safety inspection and control: A review," *Critical Reviews in Food Science and Nutrition*, vol. 52, pp. 1039–1058, 2012.
 113. M. S. Kim, A. M. Lefcourt, K. Chao, Y. R. Chen, I. Kim, and D. E. Chan, "Multispectral detection of fecal contamination on apples based on hyperspectral imagery: Part I. Application of visible and near-infrared reflectance imaging," *Transactions of the Asae*, vol. 45, 20272037, 2002.
 114. Y. Peng, and R. Lu, "Improving apple fruit firmness predictions by effective correction of multispectral scattering images," *Postharvest Biology and Technology*, vol. 41, pp. 266–274, 2006.
 115. L. Lunadei, B. Diezma, L. Lleo, L. Ruiz-Garcia, S. Cantalapiedra, and M. Ruiz-Altisent, "Monitoring of fresh-cut spinach leaves through a multispectral vision system," *Postharvest Biology and Technology*, vol. 63, pp. 74–84, 2012.
 116. F. Ma, J. Yao, T. Xie, C. Liu, W. Chen, C. Chen, L. Zheng, "Multispectral imaging for rapid and non-destructive determination of aerobic plate count (APC) in cooked pork sausages," *Food Research International*, vol. 62, pp. 902-908, 2014.
 117. B. Dissing, O. Papadopoulou, C. Tassou, B. Ersbøll, J. Carstensen, E. Panagou, and G.-J.E. Nychas, "Using Multispectral Imaging for Spoilage Detection of Pork Meat," *Food and Bioprocess Technology*, vol. 6, no. 9, pp. 2268-2279, 2013.
 118. F. Tao, Y. Peng, "A method for non-destructive prediction of pork meat quality and safety attributes by hyperspectral imaging technique," *Journal of Food Engineering*, vol. 126, pp. 98-106, 2014.
 119. F. Saadatian, L. Liu, and M. O. Ngadi, "Hyperspectral Imaging for Beef Tenderness Assessment," *International Journal of Food Processing Technology*, vol. 2, pp. 18-25, 2015.
 120. X. Sun, K. J. Chen, K. R. Maddock-Carlin, V. L. Anderson, A. N. Lepper, C. A. Schwartz, W. L. Keller, B. R. Ilse, J. D. Magolski, E. P. Berg, "Predicting beef tenderness using color and multispectral image texture features," *Meat Science*, vol. 92, no. 4, pp. 386-393, 2012.
 121. M. Kamruzzaman, D.-W. Sun, G. ElMasry, and P. Allen, "Fast detection and visualization of minced lamb meat adulteration using NIR hyperspectral imaging and multivariate image analysis," *Talanta*, vol. 103, no. 15, pp. 130-136, 2013.
 122. D. Liu, H. Pu, D.-W. Sun, L. Wang, X.-A. Zeng, "Combination of spectra and texture data of hyperspectral imaging for prediction of pH in salted meat," *Food Chemistry*, vol. 160, pp. 330-337, 2014.
 123. A. I. Ropodi, D. E. Pavlidis, F. Mohareb, E. Z. Panagou, and G.-J.E. Nychas, "Multispectral image analysis approach to detect adulteration of beef and pork in raw meats," *Food Research International*, vol. 67, pp. 12-18, Jan. 2015.
 124. M. S. Ammor, A. Argyri, and G.-J. Nychas, "Rapid monitoring of the spoilage of minced beef stored under conventionally and active packaging conditions using

- Fourier transform infrared spectroscopy in tandem with chemometrics,” *Meat Science*, vol. 81, no. 3, pp. 507-514, 2009
125. P. Skandamis, and G.-J.E. Nychas, “Preservation of fresh meat with active and modified atmosphere packaging conditions,” *Int. Journal of Food Microbiology*, vol. 79, pp. 35-45, 2002.
 126. E. Z. Panagou, O. Papadopoulou, J. M. Carstensen, and G.-J.E. Nychas, “Potential of multispectral imaging technology for rapid and non-destructive determination of the microbiological quality of beef filets during aerobic storage,” *Int. Journal of Food Microbiology*, vol. 174, pp. 1–11, 2014.
 127. B. S. Dissing, M. E. Nielsen, B. K. Ersbøll, and S. Frosch, “Multispectral imaging for determination of astaxanthin concentration in salmonids,” *PLoS ONE*, vol. 6, no. 5, 19032, 2011.
 128. S. B. Daugaard, J. Adler-Nissen, and J. M. Carstensen, “New vision technology for multidimensional quality monitoring of continuous frying of meat,” *Food Control*, vol. 21, pp. 626–632, 2010.
 129. J. G. Cruz-Castillo, S. Ganeshanandam, B. R. Mackay, G. S. Lawes, C. R. O. Lawoko, and D. J. Woolley, “Applications of canonical discriminant analysis in horticultural research,” *Hortscience*, vol. 29, no. 10, pp. 1115–1119, 1994.
 130. M. Hubert, P. Ousseuw, and K. Branden. “ROBPCA: A New Approach to Robust Principal Component Analysis,” *Technometrics*, vol. 47, no. 1, pp. 64-79, 2005.
 131. G. K. Naganathan, L. M. Grimes, J. Subbiah, C. R. Calkins, A. Samal, and G. E. Meyer, “Visible/near-infrared hyperspectral imaging for beef tenderness prediction,” *Computers and Electronics in Agriculture*, vol. 64, pp. 225–233, 2008.
 132. A. Argyri, E. Z. Panagou, P. Tarantilis, M. Polysiou, and G.-J.E. Nychas, “Rapid qualitative and quantitative detection of beef fillets spoilage based on Fourier transform infrared spectroscopy data and artificial neural networks,” *Sensors and Actuators B*, vol. 145 pp. 146-154, 2010.
 133. V. S. Kodogiannis, T. Pachidis, E. Kontogianni, “An intelligent based decision support system for the detection of meat spoilage,” *Eng. Applications of Artificial Intelligence*, vol. 34, pp. 23–36, 2014.
 134. V. S. Kodogiannis, and I. Petrounias, “Modelling of survival curves in food microbiology using adaptive fuzzy inference neural networks,” *IEEE Int. Conf. on Computational Intelligence for Measurement Systems and Applications, CIMSA*, 2012, pp. 35-40.
 135. A. Alshejari, V. S. Kodogiannis, and I. Petrounias, “An Adaptive Neuro-Fuzzy Model for the Detection of Meat Spoilage using Multispectral Images,” *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Istanbul, Turkey, 2015, pp. 1-7.
 136. S. De Jong, “SIMPLS: An alternative approach to partial least squares regression,” *Chemometrics Intell. Lab. Syst.*, vol. 18, no. 3, pp. 251-263, 1993.
 137. M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Assessment*, Wiley, 2002.

138. A. Ahmadi, M. Charwand, J. Aghaei. "Risk-constrained optimal strategy for retailer forward contract portfolio," *Int J Electr Power Energy Systems*, vol. 53, pp. 704–713, 2013.
139. R. Weron, "Electricity Price Forecasting: A Review of the State-of-the-Art with a Look into the Future," *Int. Journal of Forecasting*, vol. 30, pp. 1030–1081, 2014.
140. S. K. Aggarwal, L. M. Saini, A. Kumar. "Electricity price forecasting in deregulated markets: A review and evaluation," *Int J Electr Power Energy Systems*, vol. 31, pp. 13–22, 2009.
141. G. Li, C. Liu, C. Mattson, J. Lawarree. "Day-ahead electricity price forecasting in a grid environment," *IEEE Transactions on Power Systems*, vol. 22, pp. 266-274, 2007.
142. T. Niimura, "Forecasting Techniques for Deregulated Electricity Market Prices - Extended Survey," *Power Systems Conference and Exposition, PSCE*, 2006, pp. 51-56.
143. M. Zhou, Z. Yan, Y. X. Ni, G. Li, and Y. Nie, "Electricity Price Forecasting with Confidence-interval Estimation through an Extended ARIMA Approach," *IEE Proc.-Gener. Transm. Distrib*, vol. 153, no. 2, pp. 187-195, 2006.
144. J. Contreras, R. Espinola, F. J. Nogales, A. J. Conejo. "ARIMA Models to Predict Next-Day Electricity Prices," *IEEE Power Engineering Review*, vol. 22, no. 9, pp. 57-87, 2002.
145. A. J. Conejo, M. A. Plazas, R. Espinola, and A. B. Molina, "Day-Ahead Electricity Price Forecasting Using the Wavelet Transform and ARIMA Models," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1035-1042, 2005.
146. O. B. Fosso, A. Gjelsvik, A. Haugstad, M. Birger, and I. Wangensteen, "Generation scheduling in a deregulated system: The norwegian case," *IEEE Trans. Power Systems*, vol. 14, no. 1, pp. 75-81, 1999.
147. M. Zhou, Z. Yan, Y. Ni, and G. Li, "An ARIMA Approach to Forecasting Electricity Price with Accuracy improvement by Predicted Errors," *IEEE Power Engineering Society General Meeting*, 2004.
148. B. R. Szkuta, L. A. Sanabria, and T. S. Dillon, "Electricity Price Short-Term Forecasting using Artificial Neural Networks," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 851-857, 1999.
149. J. Guo, and P. B. Luh, "Selecting input factors for clusters of Gaussian radial basis function networks to improve market clearing price prediction," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 665–672, 2003.
150. P. Mandai, T. Senjyu, N. Urasaki, T. Funabashi, and A. K. Srivastava, "Short-Term Price Forecasting for Competitive Electricity Market," *38th North American Power Symposium*, 2006.
151. N. Amjady, and A. Daraeepour, "Day-Ahead Electricity Price Forecasting Using the Relief Algorithm and Neural Networks," *5th International Conference on European Electricity*, pp. 1-7, 2008.

152. S. Anbazhagan, and N. Kumarappan, "Day-Ahead Deregulated Electricity Market Price Forecasting Using Recurrent Neural Network," *IEEE System journal*, vol. 7, no. 4, 2013.
153. A. Mohamed, M. E. El-Hawary, "Mid-term electricity price forecasting using SVM," *IEEE Canadian Conference on Electrical and Computer Engineering*, 2016.
154. S. Fan, C. Mao, and L. Chen. "Next-day electricity-price forecasting using a hybrid network," *IET Generation, Transmission & Distribution*, vol. 1, no. 1, pp. 176-182, 2007.
155. M. J. Mahjoob, M. Abdollahzade, and R. Zarringhalam, "GA based Optimized LSSVM Forecasting of Short Term Electricity Price in Competitive Power Markets," *3rd IEEE Conference on Industrial Electronics and Applications*, Singapore, pp. 73-78, 2008.
156. Y. Y. Hong, C. Y. Hsiao, "Locational marginal price forecasting in deregulated electricity markets using artificial intelligence," *IEE Proceedings: Generation, Transmission and Distribution*, vol. 149, no. 5, pp. 621-626, 2002.
157. R. Ragavi, M. S. Kamalesh, N. Senthilnathan, "Day Ahead Electricity Price Prediction for a Distribution System in India," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 4, no. 2, pp. 669-678, 2015.
158. J. P. Catalao, H. M. Pousinho, V. M. Mendes, "Hybrid wavelet-PSO-ANFIS approach for short-term electricity prices forecasting," *IEEE Trans Power Systems*, vol. 26, no. 1 pp. 137-144, 2011.
159. H. Zhou, X. H. Wu, and G. Li, "An ANFIS model of Electricity Price Forecasting Based on Subtractive Clustering," *IEEE Power and Energy Society General Meeting*, 2011, Detroit, MI, United States, Article number 60392282011.
160. L. Xiao, W. Shao, M. Yu, J. Ma, and C. Jin, "Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting," *Applied Energy*, vol. 198, pp. 203-222, 2017
161. V. S. Kodogiannis, M. Amina, and I. Petrounias. "A clustering-based fuzzy-wavelet neural network model for short-term load forecasting," *Int. Journal of Neural Systems*, vol. 23, no. 5, 2013.
162. J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence*, Prentice-Hall, 1997.
163. I. Rojas, H. Pomares, and F. J. Fernandez, "A new methodology to obtain fuzzy systems autonomously from training data," *IEEE conf. Fuzzy System*, 1999, vol. 1, pp. 527-532.
164. V. S. Kodogiannis, T. Pachidis, and E. Kontogianni, "An intelligent based decision support system for the detection of meat spoilage," *Engineering Applications of Artificial Intelligence*, vol. 34, pp. 23-36, 2014.
165. C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller—Part I & II," *IEEE Trans. Syst. Man Cybern.SMC-20*, vol. 2, pp. 404-435, 1990.

166. O. Nelles, *Nonlinear system Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, 2001.
167. I. P. Panapakidis, and A. S. Dagoumas, “Day-ahead electricity price forecasting via the application of artificial neural network based models,” *Applied Energy*, vol. 172, pp. 132–151, 2016.
168. Q. Zhang, and A. Benveniste, “Wavelet networks,” *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 89–98, 1992.
169. M. Amina, E. Z. Panagou, V. S. Kodogiannis, G.-J.E. Nychas, “Wavelet neural networks for modelling high pressure inactivation kinetics of *Listeria monocytogenes* in UHT whole milk,” *Chemometrics and Intelligent Laboratory Systems*, vol. 103, pp. 170–183, 2010.